PAPER *Special Section on Data Engineering and Information Management*

Multi-Dimensional Fused Gromov Wasserstein Discrepancy for Edge-Attributed Graphs

Keisuke KAWANO[†a)], **Satoshi KOIDE**[†], **Hiroaki SHIOKAWA**[††], ***Nonmembers***, ***and*** **Toshiyuki AMAGASA**[††], ***Senior Member***

SUMMARY Graph dissimilarities provide a powerful and ubiquitous approach for applying machine learning algorithms to edge-attributed graphs. However, conventional optimal transport-based dissimilarities cannot handle edge-attributes. In this paper, we propose an optimal transport-based dissimilarity between graphs with edge-attributes. The proposed method, multi-dimensional fused Gromov-Wasserstein discrepancy (MFGW), naturally incorporates the mismatch of edge-attributes into the optimal transport theory. Unlike conventional optimal transport-based dissimilarities, MFGW can directly handle edge-attributes in addition to structural information of graphs. Furthermore, we propose an iterative algorithm, which can be computed on GPUs, to solve non-convex quadratic programming problems involved in MFGW. Experimentally, we demonstrate that MFGW outperforms the conventional optimal transport-based dissimilarity in several machine learning applications including supervised classification, subgraph matching, and graph barycenter calculation.

key words: *attributed graph, fused Gromov-Wasserstein divergence, graph barycenter, optimal transport*

1. Introduction

The graph is a fundamental data structure that models real-world entities and their relationships by nodes and edges, respectively. To represent complex entities, edges are usually annotated with a variety of information as edge-attributes. For example, in a chemical compound graph, atoms are represented as nodes, while bonds are represented as edges; additionally, various information such as bond types and interatomic distances are represented as edge-attributes. Such attributed graphs can be used to precisely model complex structures. Hence, it is important to handle attributed graphs for machine learning tasks such as chemical compound property prediction and crystal structure similarity search.

To handle attributed graphs in machine learning methods, it is crucial to define an appropriate dissimilarity between graphs. Given two attributed graphs, dissimilarities quantify the differences between them by considering their structures and attributes. Designing dissimilarities allows the application of machine learning methods such as kernels [1] and t-distributed stochastic neighborhood embedding (t-SNE) [2]. Fused Gromov-Wasserstein discrepancy (FGW) has been proposed as one of the representative graph dissimilarities [1], [3], [4]. FGW measures dissimilarity by combining two optimal transport-based (OT) dissimilarities [5], [6], considering differences in both structures and node-attributes. To compute FGW, a quadratic programming problem needs to be solved, and an iterative algorithm has been proposed to obtain local stationary points in a reasonable time [1], [7], [8]. As reported in the previous studies [1], the kernels obtained by FGW have shown better performance in machine learning tasks such as graph classification than conventional kernels including shortest path kernel [9], random walk kernel [10], [11], and Wasserstein Weisfeiler-Lehman kernel [12].

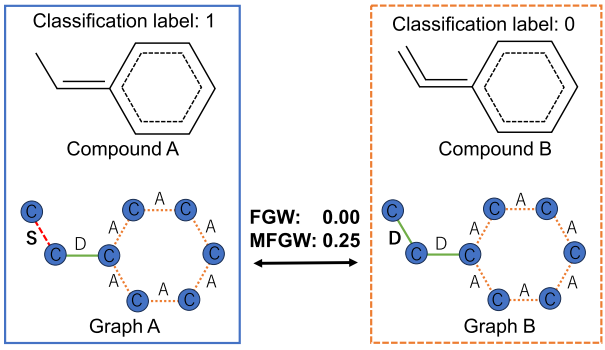


Fig. 1 Examples of chemical compounds and their corresponding graphs: In the graphs, nodes and edges are corresponding to atoms and chemical bonds, respectively. Each node has a node-label, which represents atom type (C). Each edge has an edge-label, which represents one of the bond types among single bond (S), double bond (D), and aromatic bond (A). The graph has a classification label 0 or 1, which represents the carcinogenicity of the compound.

Although FGW is effective in many graph-based applications, it has two critical weaknesses. The first is that FGW cannot handle edge-attributes such as bond types, interatomic distances, etc. These attribute values must be cut off before using FGW, resulting in a negative impact on the performance of the downstream task. The second is that the existing iterative algorithm for FGW [1], [3], [4] tends to converge to a local optimum solution even though it is significantly far from the global optimum. As a result, dissimilarity accuracy is reduced, causing low performance of subsequent machine learning tasks.

Figure 1 illustrates practical chemical compounds included in PTC MR dataset [13] and their corresponding graph representations. The compounds A and B have differ-

Manuscript received June 27, 2023.
Manuscript revised November 15, 2023.
Manuscript publicized January 12, 2024.

[†]The authors are with the Toyota Central R&D Labs., Inc., Nagakute-shi, 480–1192 Japan.
[††]The authors are with the Center for Computational Sciences, University of Tsukuba, Tsukuba-shi, 305–8577 Japan.
a) E-mail: kawano@mosk.tytlabs.co.jp

DOI: 10.1587/transinf.2023DAP0014

ent substructures and classification labels. However, FGW yields 0.0 between A and B, which means that FGW cannot distinguish the compounds. This is because FGW needs to cut off the edge-attributes. As shown in the figure, the graphs A and B have the same structural information and attributes except for edge-attributes. Thus, the kernels based on FGW fail to disambiguate the graphs, resulting in poor classification quality if we use such kernels[†].

We propose multi-dimensional fused Gromov Wasserstein discrepancy (MFGW) for attributed-graphs. To overcome the aforementioned limitations, we design a novel objective function by extending FGW so that MFGW can directly measure the difference between the edge-attributes. Unlike the existing methods, MFGW can handle edge-attributes, which are crucial for many machine learning tasks. In addition, we propose an iterative optimization method with heuristics to avoid the local optimum solution, resulting in a higher performance in downstream tasks. Our experiments show that MFGW outperforms FGW in several graph machine learning tasks, including graph classification over edge-attributed graphs. Our contributions can be summarized as follows:

1. We propose an OT dissimilarity, namely MFGW, for attributed graphs (Sect. 4). Unlike existing methods [1], [3], MFGW can directly handle edge-attributes.
2. We propose a heuristic optimization method for the quadratic programming involved in OT dissimilarities (Sect. 4.2). By using our method, MFGW achieves better performance than existing OT dissimilarities.
3. We theoretically clarify that MFGW is a pseudometric (Sect. 4.3), which indicates that MFGW is applicable to various machine learning algorithms.

Our experiments show that MFGW is applicable to fundamental machine learning tasks, graph classification, subgraph matching, and gradient-based optimization. For example, in the classification, MFGW outperformed the other graph kernel methods on the datasets containing edge-attributed graphs. Furthermore, our experimental results demonstrated that the proposed iterative optimization successfully mitigates the local optimum solution problem.

The remainder of this paper is as follows. Section 2 provides a literature review of related work. Section 3 introduces the optimal transport on graphs. In Sect. 4, MFGW and the optimization method are proposed. Section 5 presents experimental results, and Sect. 7 concludes the paper.

2. Related Work

In Sect. 2.1, we first discuss general OT dissimilarities such as Wasserstein distance, that is followed by OT dissimilarities for graphs in Sect. 2.2. After that, we discuss about graph kernels and subgraph matching in Sect. 2.3.

[†]With FGW between the line graphs, the distance between graphs can be obtained by considering edge-attributes instead of node-attributes. However, FGW cannot consider both node- and edge-attributes simultaneously.

2.1 OT Dissimilarities for Distributions

OT dissimilarities are one of the popular metrics that quantify the difference between variable-length and order-invariant data such as data distributions [5], [14]. To measure the difference between two data distributions, OT dissimilarity finds the optimal transport from the first distribution to the second one; the optimal transport is the lowest cost way of moving data points between the distributions. In the machine learning community, Wasserstein distance [5] and Gromov-Wasserstein discrepancy (GW) [15] are the most prevalent OT dissimilarities. Wasserstein distance finds the optimal transport with pairwise distances between data points in the two distributions within the same metric space. In contrast, GW first calculates the distances between pairs of data points within each distribution. Then, it calculates the pairwise distances between the calculated distances. Finally, GW finds the optimal transport using these pairwise distances. These OT dissimilarities formulate a linear programming problem and a quadratic programming problem, respectively.

2.2 OT Dissimilarities for Graphs

In recent years, the above OT dissimilarities are regarded as an essential tool to compare the differences between graphs [1], [3], [4], [16]. Unlike the data distribution handled in the above OT dissimilarities, a graph has two types of information: structural information of nodes and edges, and attribute information included in the graph. However, the above OT dissimilarities do not handle the structural and attribute information. Thus, several extensions of the OT dissimilarities have been proposed so as to effectively quantify the difference between graphs.

One promising approach is the Wasserstein Weisfeiler-Lehman kernel (WWL) [16]. WWL first transforms the structural information of a graph into WL features [12]. It then quantifies dissimilarity between two WL features by using Wasserstein distance. Similarly, Wasserstein embedding for graph learning extends WWL by using the message passing scheme to generate the WL features [17].

Although WWL and its extensions are applicable for graphs, they cannot distinguish the differences among several classes of graphs [18]. This is because that the structural information represented by WL feature is incomplete. To overcome this limitation, fused Gromov Wasserstein discrepancy (FGW) has recently been proposed [1], [3], [4]. FGW integrates GW and Wasserstein distance to measure the differences in both structural information and node-attributes; it uses GW and Wasserstein distances to measure the structural information and the node-attributes, respectively. However, as discussed in Sect. 1, FGW still has a critical weakness: FGW cannot measure the differences in the edge-attributes even though real-world graphs typically have edge-attributes such as bond types (Fig. 1). Several extensions of FGW have been proposed [3], [19], but there is still an open problem in handling edge-attributes.

To overcome the above limitations, we propose a novel OT dissimilarity, namely MFGW, for edge-attributed graphs. Recall that the aforementioned OT dissimilarities cannot handle edge-attributes to quantify the difference between two graphs. By contrast, MFGW is designed to integrate the structural information and the edge-attributes. Thus, MFGW can use full information on graphs, which is crucial for various machine learning tasks using attributed graphs.

2.3 Graph Kernels and Subgraph Matching

To evaluate similarities among graphs, graph kernels and subgraph matching are ones of the most traditional approaches. Herein, we discuss relationships between MFGW and such traditional approaches. A graph kernel is a kernel function that measures the similarity of pairs of graphs, which is mainly used for graph classification tasks. Given a pair of graphs, a graph kernel measures the similarity based on features extracted from the graphs such as WL graph kernel [12], FGW kernel [1], and the shortest path distance kernel [9]. For example, in the shortest path distance kernel, a graph kernel measures the similarity by comparing the distributions of shortest path distances and node attributes in two graphs. However, unlike MFGW, these kernel functions cannot handle attributes associated with nodes and edges in graphs. To overcome this limitation, the deep divergence graph kernel (DDGK) [20] has recently proposed. To cope with the attributes, DDGK builds neural network based autoencoders by using a training graph dataset. However, as we will show in Sect. 5.1, DDGK degrades its accuracy if the database is small since it is difficult for DDGK to tune its hyperparameters on small datasets. By contrast, MFGW shows better performance even if the training data is small because MFGW contains only a small amount of parameters (α and β). In Sect. 5.1, we will experimentally discuss how MFGW works well compared to DDGK on real-world datasets.

A subgraph matching is a traditional graph search approach that extracts subgraphs which are isomorphic or quasi-isomorphic to a query graph from a database. Recently, several algorithms have been proposed to handle node- and edge-attributed graphs [21], [22]. For instance, Du et al. [22] proposed an interactive subgraph matching algorithm for attributed graphs. However, existing algorithms considers only discrete labels for the attributes, and they thus cannot handle attributes composed of continuous values. One way to overcome this limitation is thresholding of the difference between continuous edge-attributes. Cordella et al. proposed VF2 [23] that finds matched edge pairs by thresholding the difference between continuous edge-attributes, and then finds subgraph matching based on the matched edges. However, the search quality of VF2 is a quite sensitive to the thresholds, and it is difficult to specify an effective thresholds if graphs have diverse attribute values. By contrast, MFGW can handle not only discrete labels but also continuous values as for the node- and edge-attributes without using user-specified thresholds. As a result, as we will show in Sect. 5.2, MFGW can yield more flexible and reasonable searches compared to VF2.

3. Preliminary

First, the basic notations are defined, followed by a brief overview of OT dissimilarities on graphs [1], [3], [4], [16].

3.1 Notations

A vector and a matrix are denoted in bold style (e.g., $\mathbf{x}$ and $\mathbf{A}$), and x_i and A_{ij} represent the i-th and (i,j)-th elements, respectively. $\mathbf{1}_N \in \mathbb{R}^N$ denotes an N-dimensional vector with all elements as ones. An undirected graph is denoted by $G = (V, E, w, f, g)$, where V and E are the sets of nodes and edges, respectively. $w : E \mapsto \mathbb{R}$ is a function that maps edges to their corresponding weights. $f : V \mapsto \mathbb{R}^{d_f}$ and $g : E \mapsto \mathbb{R}^{d_g}$ are functions that respectively map nodes and edges to their corresponding attributes, where d_f and d_g are the dimensions of the attributes. For convenience, we denote $N = |V|$ and $\mathbf{x}_i = f(i) \in \mathbb{R}^{d_f}$.

3.2 FGW

The OT dissimilarities are defined as the minimum cost of transporting masses virtually assigned to the nodes. Given two graphs G and $\tilde{G}$, FGW [1], [3], [4] is defined as

$$\mathrm{FGW}(G, \tilde{G}) = \left(\min_{\mathbf{P} \in \mathcal{P}} \sum_{i,j,k,l} \left(L_{ijkl}\right)^p P_{ik} P_{jl} \right)^{\frac{1}{p}}, \tag{1}$$

$$L_{ijkl} = \alpha \left\| \mathbf{x}_i - \tilde{\mathbf{x}}_k \right\|_q^q + (1-\alpha) \left| C_{ij} - \tilde{C}_{kl} \right|^q, \tag{2}$$

where $\mathbf{P} \in [0,1]^{N \times \tilde{N}}$ is a correspondence matrix whose element $P_{ik} \in [0,1]$ represents a mass transported from node i in G to node k in $\tilde{G}$. Recall that $\mathbf{x}_i$ and $\tilde{\mathbf{x}}_k$ are the node-attribute vectors of node i in G and node k in $\tilde{G}$, respectively. In the literature [1], [16], node-labels and WL features are used to build the node-attribute vectors. C_{ij} and $\tilde{C}_{kl} \in \mathbb{R}$ represent intra-graph similarities that evaluate how two adjacent nodes are strongly related. Specifically, C_{ij} denotes the similarity between nodes i and j in G, where the similarity is measured by using the edge-weight $w(i,j)$ or the shortest path distance from node i to j [1]. $\alpha \in [0,1]$ is a parameter to balance the importance of the node-attributes and the intra-graph similarities and $p, q \geq 1$. WWL [16] can be regarded as a special case of FGW, where $\alpha = 1$. In Eq. (1), $\mathcal{P}$ is a set of the valid correspondence matrices satisfying the following constraints: **(C1)** $\mathbf{P}\mathbf{1}_{\tilde{N}} \leq \mathbf{a}$, **(C2)** $\mathbf{P}^\top \mathbf{1}_N \leq \tilde{\mathbf{a}}$, and **(C3)** $\mathbf{1}_N^\top \mathbf{P} \mathbf{1}_{\tilde{N}} = \min(\mathbf{a}^\top \mathbf{1}_N, \tilde{\mathbf{a}}^\top \mathbf{1}_{\tilde{N}})$, where $\mathbf{a} \in [0,1]^N$ and $\tilde{\mathbf{a}} \in [0,1]^{\tilde{N}}$ are the masses virtually assigned to the nodes in G and $\tilde{G}$, respectively. (C1) and (C2) respectively indicate that the transported mass should be no more than $\mathbf{a}$ and $\tilde{\mathbf{a}}$. (C3) indicates that the total amount of transported mass needs to be equivalent to the smaller one of the total masses.

As shown in Eq. (1), FGW is regarded as a non-convex optimization problem [1], which can be solved by using a

quadratic programming (QP) solver. However, the QP solver incurs considerable computational costs if the matrix $\mathbf{P}$ is large. To reduce this cost, Vayer et al., proposed an iterative method based on conditional gradients to guarantee convergence to a local solution [1]. The algorithm repeatedly computes the gradient of L_{ijkl} with respect to $\mathbf{P}$, and it updates $\mathbf{P}$ by performing the network flow algorithm [14].

4. Proposed Method

In this section, we propose the multi-dimensional fused Gromov Wasserstein discrepancy (MFGW), an OT dissimilarity between edge-attributed graphs. We first formulate MFGW as a quadratic programming problem in Sect. 4.1, followed by the heuristic optimization algorithm to solve the problem in Sect. 4.2. We then theoretically demonstrate that MFGW is a pseudometric in Sect. 4.3, and we finally discuss the applications of MFGW in Sect. 4.4.

4.1 MFGW

Similar to FGW, MFGW is the OT dissimilarity, which is regarded as the minimum cost of transporting masses. Given two attributed graphs $G = (V, E, w, f, g)$ and $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{w}, \tilde{f}, \tilde{g})$, MFGW is formally defined as follows:

$$\mathrm{MFGW}(G, \tilde{G}) = \left(\min_{\mathbf{P} \in \mathcal{P}} \sum_{i,j,k,l} L_{ijkl}^{(\mathrm{MFGW})} P_{ik} P_{jl} \right)^{\frac{1}{p}}, \tag{3}$$

$$L_{ijkl}^{(\mathrm{MFGW})} = \alpha \|\mathbf{x}_i - \tilde{\mathbf{x}}_k\|_p^p + (1-\alpha) \|\mathbf{e}_{ij} - \tilde{\mathbf{e}}_{kl}\|_{p,\beta}^p, \tag{4}$$

where $p \geq 1$ and $\alpha \in [0, 1]$. Unlike FGW, MFGW measures the difference between $\mathbf{e}_{ij}$ and $\tilde{\mathbf{e}}_{kl} \in \mathbb{R}^{d_g+1}$ instead of C_{ij} and $\tilde{C}_{kl}$, where $\mathbf{e}_{ij}$ and $\tilde{\mathbf{e}}_{kl}$ are the edge-embedding vectors. Formally, each element of $\mathbf{e}_{ij}$ is defined as

$$\left(\mathbf{e}_{ij}\right)_m = \begin{cases} C_{ij} & (m = 1), \\ (g(i,j))_{m-1} & (m > 1). \end{cases} \tag{5}$$

In Eq. (5), $g(i, j)$ is the edge-attribute vector on edge (i, j), e.g., one-hot vectors of edge-labels, and C_{ij} denotes the intra-graph similarities. $\|\mathbf{e}_{ij} - \tilde{\mathbf{e}}_{kl}\|_{p,\beta}^p = \beta |C_{ij} - \tilde{C}_{kl}|^p + (1 - \beta) \|g(i,j) - \tilde{g}(k,l)\|_p^p$ is a weighted p-norm, where $\beta \in [0, 1]$ is a parameter to balance the importance of the intra-graph similarities and edge-attribute vectors. If we set $\beta = 1$, the objective function of MFGW is equivalent to that of FGW. As shown in Eq. (2), FGW considers only the intra-graph similarity C_{ij}. By contrast, MFGW handles not only the structural similarity but also the edge-attributes $g(i, j)$. This extension allows edge-attributes to be processed in the OT dissimilarities without dropping off all attribute values included in real-world graphs.

Figure 2 illustrates how to concretely compute MFGW. For simplicity, we assume all masses for the constraints (C1)-(C3) are the same, i.e., $\mathbf{a}_i = \tilde{\mathbf{a}}_k = \frac{1}{4}$ for all $i, k \in \{1, 2, 3, 4\}$. Also, all nodes have the same node-attribute "0" in order to focus only on the differences in the structural information and edge-attributes. As shown in the figure, the edges are labeled with either "1" or "2", which correspond to the line style. If the shortest path distances are employed to measure the structural information, the intra-graph similarities are computed as follows, where $[\![C_{ij}]\!]$ denotes the matrix with C_{ij} as the (i, j)-th element.

$$[\![C_{ij}]\!] = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix}, \quad [\![\tilde{C}_{kl}]\!] = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix}. \tag{6}$$

$g(i, j)$ is corresponding to the edge-label. By letting $g(i, j) = 0$ be the absence of an edge between nodes i and j, we have the following matrices for the edge-attributes. For the simplicity, we denote $[\![g_{ij}]\!] = [\![g(i, j)]\!]$.

$$[\![g_{ij}]\!] = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad [\![\tilde{g}_{kl}]\!] = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 2 \\ 0 & 1 & 2 & 0 \end{pmatrix}. \tag{7}$$

As shown above, Eqs. (6) and (7) encode the structural information and the edge-attributes, respectively. MFGW finds the optimal transport using the edge-embeddings described in Eq. (5), which contain both types of information. For example, we have $\mathbf{e}_{2,4} = [1, 1]^\top$ and $\tilde{\mathbf{e}}_{3,4} = [1, 2]^\top$, where the values in the first dimension are $C_{2,4}$ and $\tilde{C}_{3,4}$, while the values in the second dimension are $g_{2,4}$ and $\tilde{g}_{3,4}$.

By solving the optimization problem in Eq. (3), we can obtain the MFGW value and the correspondence matrix. For example, if $\alpha = 0$, we can derive $\mathrm{MFGW}(G, \tilde{G}) = 0.125$, and its correspondence matrix is

$$\mathbf{P} = \begin{pmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{4} \end{pmatrix}, \tag{8}$$

which represents the transported masses from the nodes in G to the nodes in $\tilde{G}$.

The transported mass can be regarded as the alignment between the nodes. MFGW aligns the nodes in the graphs, and it quantifies the dissimilarity by considering the differences in both the structural information and the edge-attributes. For instance, Fig. 2, the nodes 2 and 4 in G are respectively aligned to the nodes 3 and 4 in $\tilde{G}$ by following $\mathbf{P}$ shown in Eq. (8). As shown in Eq. (3), MFGW quantifies the difference between $g(2, 4)$ and $\tilde{g}(3, 4)$. In contrast, FGW is unable to handle edge-attributes based on Eq. (4). Thus, $\mathrm{FGW}(G, \tilde{G})$ should be 0.0 even though G and $\tilde{G}$ have different edge-labels in edges $(2, 4)$ and $(3, 4)$, respectively.

Furthermore, it is important to note that $\mathrm{FGW}(G, G') = \mathrm{FGW}(\tilde{G}, G')$ holds for any graphs G' if we have structurally similar graphs G and $\tilde{G}$ like the ones shown in Fig. 2. This implies that it is difficult for FGW to discriminate two different graphs G and $\tilde{G}$ by using FGW kernel [1]. By contrast,

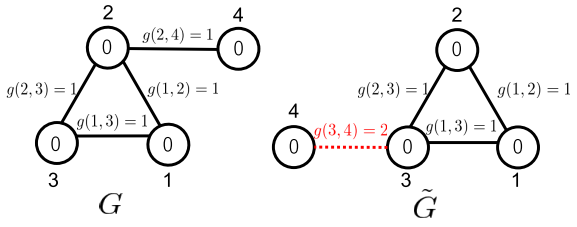


Fig. 2 Example of edge-attributed graphs. The numbers in nodes represent node-attributes. The numbers on edges and line styles represent edge-labels. The remaining numbers represent indices of the nodes. The graphs are isomorphic except for the edge-labels between nodes, $g(2,4)$ and $\tilde{g}(3,4)$.

MFGW$(G,G') \neq$ MFGW$(\tilde{G},G')$ holds for any G' except for the special graphs (i.e., midpoints between G and $\tilde{G}$) since MFGW can evaluate the differences between G and $\tilde{G}$ by handling the edge-attributes. Hence, unlike the FGW kernel, the kernel based on MFGW does not fail to discriminate the two different graphs G and $\tilde{G}$.

4.2 Optimization for MFGW

Similar to FGW, MFGW defined in Eq. (3) is clearly a quadratic programming problem. We propose an iterative algorithm to solve the problem by using the Bregman proximal gradient descent [24]. We designed the algorithm as a parallel computation manner. Algorithm 1 shows the overview of the iterative algorithm, which is inspired by [25] and [3]. At the beginning of the optimization, the correspondence matrix is uniformly initialized i.e., $\mathbf{P}^{(0)} = \mathbf{a}\tilde{\mathbf{a}}^\top$ (line 1). In step $t+1$, the correspondence matrix $\mathbf{P}^{(t+1)}$ is updated as follows (lines 2-7):

$$\mathbf{P}^{(t+1)} = \underset{\mathbf{P}\in\mathcal{P}}{\operatorname{argmin}} \frac{1}{\epsilon}\left\langle \mathbf{D}^{(t)}, \mathbf{P}\right\rangle + d_{\mathrm{KL}}(\mathbf{P}\|\mathbf{P}^{(t)}), \tag{9}$$

$$\mathbf{D}^{(t)} = \nabla\left\{\sum_{i,j,k,l} L_{ijkl}^{(\mathrm{MFGW})} P_{ik}P_{jl}\right\}\Bigg|_{P=P^{(t)}}, \tag{10}$$

where $\langle\cdot,\cdot\rangle$ is inner-product of matrix, $d_{\mathrm{KL}}(\mathbf{P}\|\mathbf{P}^{(t)}) = \sum_{j,l} P_{jl}\log\frac{P_{jl}}{P_{jl}^{(t)}} - P_{jl} + P_{jl}^{(t)}$ is the KL-divergence, and $\epsilon > 0$ is the inverse step size. Consequently, Eqs. (9) and (10) can be reformulated as follows:

$$\mathbf{P}^{(t+1)} = \underset{\mathbf{P}\in\mathcal{P}}{\operatorname{argmin}} \sum_{i,k}(D_{ik}^{(t)} - \epsilon\log P_{ik}^{(t)} + \epsilon)P_{ik} - \epsilon H(\mathbf{P}), \tag{11}$$

$$D_{ik}^{(t)} = \alpha\|\mathbf{x}_i - \tilde{\mathbf{x}}_k\|_p^p + 2(1-\alpha)\sum_{j,l}\|\mathbf{e}_{ij} - \tilde{\mathbf{e}}_{kl}\|_{p,\beta}^p P_{jl}^{(t)}, \tag{12}$$

where $H(\mathbf{P})$ is an entropy of $\mathbf{P}$. The optimization problem shown in Eq. (11) is known as an entropy-regularized optimal transport problem, which can be solved by the generalized Sinkhorn iteration [5] on GPUs. The computational complexity for each step (Eqs. (11) and (12)) is $O(N^2\tilde{N}^2)$ time, where N and $\tilde{N}$ are the number of nodes in G and $\tilde{G}$, respectively. This is the same complexity as FGW [1]. If $p=2$, the computational complexity of MFGW becomes $O(N^2\tilde{N}+N\tilde{N}^2)$ time, which is the same cost as [1] and [26].

Algorithm 1 MFGW

Input: $G, \tilde{G}, \epsilon_{\mathrm{init}}, \epsilon_{\mathrm{target}}, T$
Output: $\mathbf{P}^{(T)}$, MFGW$(G,\tilde{G})$
1: $\epsilon \leftarrow \epsilon_{\mathrm{init}}$, $\mathbf{P}^{(0)} \leftarrow \mathbf{a}\tilde{\mathbf{a}}^\top$
2: **for** $t = 0 : T-1$ **do**
3: Obtain $\mathbf{D}^{(t)}$ by Eq. (12).
4: Add noises to $\mathbf{D}^{(t)}$ by Eq. (13).
5: Update $\mathbf{P}^{(t+1)}$ by Eq. (11) based on Sinkhorn iteration.
6: $\epsilon \leftarrow \epsilon_{\mathrm{init}} + \frac{t}{T}(\epsilon_{\mathrm{target}} - \epsilon_{\mathrm{init}})$
7: **end for**
8: MFGW$(G,\tilde{G}) \leftarrow (\sum_{i,j,k,l} L_{ijkl}^{(\mathrm{MFGW})} P_{ik}^{(T)} P_{jl}^{(T)})^{\frac{1}{p}}$

Similar to FGW, Eqs. (11) and (12) may lead to converged matrix $\mathbf{P}^{(T)}$ that is a local minimum solution. This is because that the iterative algorithm starts from an initial matrix that is far from the optimal solution even though MFGW is non-convex. To alleviate this problem, we propose a deterministic annealing approach. At the beginning of the iterations ($t=0$), ϵ is set to a sufficiently large value ϵ_{init}, and as the iteration proceeds, ϵ is gradually reduced to the target value $\epsilon_{\mathrm{target}}$. Intuitively, at the initial iteration, the objective function is convex. Its optimal solution is the correspondence matrix with the maximum entropy, i.e., $\mathbf{P}^{(0)} = \mathbf{a}\tilde{\mathbf{a}}^\top$. As the iteration progresses, the objective function gradually transforms back to its original non-convex form by reducing ϵ. As the objective function changes, the optimal solution also gradually changes from $\mathbf{P}^{(0)}$ to the original one. By updating the correspondence matrix along with the changes, we can start the optimization from the close point to the optimal solution in each step. Thus, the deterministic annealing alleviates the problem of the local optimal solutions.

Besides the above approach, we further extend Eq. (12) so as to avoid the saddle point of the objective function in Eq. (3). Specifically, as shown in line 4 in Algorithm 1, we add a small amount of noises $\Delta_{D_{ik}}$ into Eq. (12) as follows:

$$D_{ik}^{(t)} \leftarrow \mathrm{ReLU}(D_{ik}^{(t)} + \Delta_{D_{ik}}),\ \ \Delta_{D_{ik}} \sim \mathcal{N}(0,\sigma), \tag{13}$$

where $\mathcal{N}$ is the Gauss distribution and $\sigma > 0$ is a small variance. Eq. (13) employs ReLU function in order to maintain $D_{ik}^{(t)} \geq 0$. The saddle points are caused by equivalent solutions in the optimal transport problem. For example, Fig. 2 has two equivalent solutions; the nodes $1,2,3$, and 4 in G can be respectively corresponding to both the nodes $1,3,2$, and 4 and the nodes $2,3,1$, and 4 in $\tilde{G}$. By adding noise, it is possible to choose one of these solutions in order to avoid convergence to the saddle point.

4.3 A Pseudometric Property of MFGW

MFGW is a pseudometric, which is an important property to use MFGW in various machine learning methods. To prove this property, we first theoretically verify that MFGW always satisfies the following properties:

Proposition 1. *MFGW satisfies the following properties for*

any graph pairs $(G, \tilde{G})$.

1. $MFGW(G, \tilde{G}) \geq 0$
2. $MFGW(G, G) = 0$
3. $MFGW(G, \tilde{G}) = MFGW(\tilde{G}, G)$

Proof. For a pair of the graphs, G and $\tilde{G}$, $\mathbf{P}^*(G, \tilde{G})$ denotes the optimal correspondence matrix obtained from Eq. (3). By using this matrix, we can prove the properties as follows:

1. From the definition, $\mathbf{P}^*(G, \tilde{G}) \in [0, 1]^{N \times \tilde{N}}$ holds, where N and $\tilde{N}$ are the number of nodes in G and $\tilde{G}$, respectively. p-norm is always greater than or equal 0. Thus, $\mathrm{MFGW}(G, \tilde{G}) \geq 0$ from Eq. (3).
2. We assume a correspondence matrix $\hat{\mathbf{P}} = [\![\delta_{ij} a_i]\!] \in [0, 1]^{N \times N}$, where a_i is the mass on the i-th node and δ_{ij} is the Krocker delta. Since $\|\mathbf{x}_i - \mathbf{x}_i\|_p^p = 0$ and $\|\mathbf{e}_{ij} - \mathbf{e}_{ij}\|_{p,\beta}^p = 0$ hold for any i, j, the objective function in Eq. (3) become 0 at $\mathbf{P} = \hat{\mathbf{P}}$. Since MFGW is non-negative, $\mathbf{P}^*(G, G) = \hat{\mathbf{P}}$ and $\mathrm{MFGW}(G, G) = 0$ hold.
3. Since p-norm is symmetric, $\mathbf{P}^*(G, \tilde{G}) = \left(\mathbf{P}^*(G, \tilde{G})\right)^\top$ hold from Eqs. (3) and (4). Thus, $\mathrm{MFGW}(G, \tilde{G}) = \mathrm{MFGW}(\tilde{G}, G)$ holds from Eq. (3). □

In addition to Proposition 1, MFGW satisfies the following triangle inequality if total masses are the same:

Proposition 2. *Let* $\{G, \tilde{G}, \hat{G}\}$ *be a graph set and* $\mathbf{a}, \tilde{\mathbf{a}}, \hat{\mathbf{a}}$ *be the masses of the nodes in graphs, respectively. For any graph set* $\{G, \tilde{G}, \hat{G}\}$, $MFGW(G, \tilde{G}) \leq MFGW(G, \hat{G}) + MFGW(\hat{G}, \tilde{G})$ *holds, if* $\mathbf{a}\mathbf{1}_N = \tilde{\mathbf{a}}\mathbf{1}_{\tilde{N}} = \hat{\mathbf{a}}\mathbf{1}_{\hat{N}} = 1$, *where* $N, \tilde{N}, \hat{N}$ *are the number of nodes in* $G, \tilde{G}, \hat{G}$, *respectively.*

Proof. For a pair of the graphs $(G, \tilde{G})$, $\mathbf{P}^{(G,\tilde{G})}$ denotes the optimal correspondence matrix obtained from Eq. (3). Let $\mathbf{S} = \mathbf{P}^{(G,\hat{G})} \mathrm{diag}(1/\mathbf{a}') \mathbf{P}^{(\hat{G},\tilde{G})}$ be a feasible correspondence matrix between G and $\tilde{G}$. $\mathrm{diag}(\mathbf{x})$ denotes a diagonal matrix, where the diagonal elements are $\mathbf{x}$. Each element of $\mathbf{a}'$ is defined as $a'_i = \hat{a}_i$ if $\hat{a}_i \neq 0$; otherwise, $a'_i = 1$. We also denote the costs in Eq. (4) for between G and $\tilde{G}$ as $L_{ijkl}^{(G,\tilde{G})}$. Since $\mathbf{S}$ is a feasible solution but not necessarily an optimal solution, the following inequality hold.

$$\mathrm{MFGW}(G, \tilde{G}) \leq \left(\sum_{ijkl} L_{ijkl}^{(G,\tilde{G})} S_{ik} S_{jl} \right)^{\frac{1}{p}} \tag{14}$$

$$= \left(\sum_{ijklmn} L_{ijmn}^{(G,\tilde{G})} \frac{P_{im}^{(G,\hat{G})} P_{mk}^{(\hat{G},\tilde{G})} P_{jn}^{(G,\hat{G})} P_{nl}^{(\hat{G},\tilde{G})}}{a'_m a'_n} \right)^{\frac{1}{p}}. \tag{15}$$

Since we can regard $L_{ijkl}^{(G,\tilde{G})}$ as weighted p-norm between the vectors, $\left[\alpha^{\frac{1}{p}} \mathbf{x}_i^\top, (1-\alpha)^{\frac{1}{p}} \mathbf{e}_{ij}^\top\right]^\top$ and $\left[\alpha^{\frac{1}{p}} \tilde{\mathbf{x}}_k^\top, (1-\alpha)^{\frac{1}{p}} \tilde{\mathbf{e}}_{kl}^\top\right]^\top$, we have the following from Minkowski's inequality [27].

$$\mathrm{MFGW}(G, \tilde{G})$$

$$\leq \left(\sum_{ijklmn} L_{ijmn}^{(G,\hat{G})} \frac{P_{im}^{(G,\hat{G})} P_{mk}^{(\hat{G},\tilde{G})} P_{jn}^{(G,\hat{G})} P_{nl}^{(\hat{G},\tilde{G})}}{a'_m a'_n} \right)^{\frac{1}{p}}$$

$$+ \left(\sum_{ijklmn} L_{mnkl}^{(\hat{G},\tilde{G})} \frac{P_{im}^{(G,\hat{G})} P_{mk}^{(\hat{G},\tilde{G})} P_{jn}^{(G,\hat{G})} P_{nl}^{(\hat{G},\tilde{G})}}{a'_m a'_n} \right)^{\frac{1}{p}}. \tag{16}$$

Since $\mathbf{a}^\top \mathbf{1}_N = \tilde{\mathbf{a}}^\top \mathbf{1}_{\tilde{N}} = \hat{\mathbf{a}}^\top \mathbf{1}_{\hat{N}} = 1$ and the constraints (C1), (C2), and (C3), we have $\sum_i P_{im}^{(G,\hat{G})} = \sum_k P_{mk}^{(\hat{G},\tilde{G})} = a'_m$ and $\sum_j P_{jn}^{(G,\hat{G})} = \sum_l P_{nl}^{(\hat{G},\tilde{G})} = a'_n$. Thus, we have $\mathrm{MFGW}(G, \tilde{G}) \leq \mathrm{MFGW}(G, \hat{G}) + \mathrm{MFGW}(\hat{G}, \tilde{G})$. Hence, from the above inequalities, Proposition 2 holds. □

Propositions 1 and 2 indicate that MFGW is a pseudo-metric only if the condition of Proposition 2 holds. The condition means a "non-partial" setting such that all the masses should be transported, which is a commonly used setting in the Wasserstein distance and FGW [1]. Thus, as well as the existing studies [1], [12], [16], MFGW is applicable to various existing machine learning techniques such as k nearest neighbor classifications [28] and t-SNE [2].

4.4 Applications

Finally, we discuss how MFGW is applicable for machine learning tasks. In addition to directly measuring the dissimilarities, we can utilize MFGW in the following applications:

(1) Indefinite Kernel

MFGW can be used for classification tasks [28] by constructing the indefinite kernel as $k(G, \tilde{G}) = \exp(-\lambda \mathrm{MFGW}(G, \tilde{G}))$, where λ denotes the parameter such that $\lambda > 0$. By constructing the kernel function, it is possible to incorporate MFGW into classification methods.

(2) Subgraph Matching

Given a query graph G_q and a target graph G_t, the task of finding a subgraph included in G_t that matches G_q is called subgraph matching [23]. If the masses on the nodes are the same for all nodes (i.e., $a_i = \tilde{a}_j$), MFGW can be regarded as a continuous relaxation of subgraph matching. Once the MFGW between the query and the target graph is calculated, the correspondence matrix $\mathbf{P}^*$, which is the solution of the optimization problem in Eq. (3), can be obtained. This correspondence matrix can be used as the solution of subgraph matching because $\mathbf{P}^*$ represents the nodes in the target graph that match the nodes in the query. Unlike FGW [1], [3], [29], MFGW performs subgraph matching by considering edge-attributes. Note that MFGW does not require any additional parameters (e.g., thresholds for node and edge matching) even if attributes do not strictly match, because MFGW finds the subgraph by minimizing the sum of the transportation costs.

(3) Gradient of MFGW

Some machine learning applications require gradients with

respect to the inputs. The gradient of MFGW with respect to node-attributes or edge-embeddings can be obtained in two ways. The first way is the algorithmic gradient with the autograd technique [30], because the iterations in Eqs. (11) and (12) and the generalized Sinkhorn iteration are algorithmically differentiable. Although algorithmic gradients can be used even if iterations do not converge sufficiently [14], [31], it is necessary to store the correspondence matrices in each iteration, which requires a large memory footprint. The second way is an approximation using the correspondence matrix in the last step $\mathbf{P}^{(T)}$. Given an optimal solution $\mathbf{P}^*$, the gradient of MFGW with respect to the edge-embedding $\mathbf{e}_{ij}$ can be obtained as follows:

$$\frac{\partial \mathrm{MFGW}(G,\tilde{G})^p}{\partial \mathbf{e}_{ij}} = \sum_{k,l}(1-\alpha)P^*_{ik}P^*_{jl}\frac{\partial \|\mathbf{e}_{ij}-\tilde{\mathbf{e}}_{kl}\|^p_{p,\beta}}{\partial \mathbf{e}_{ij}}. \quad (17)$$

The gradient of MFGW can be estimated using $\mathbf{P}^{(T)}$ instead of $\mathbf{P}^*$. The gradient with respect to the node-attributes $\mathbf{x}_i$ can be obtained in the same manner. In our experiments, we employ Eq. (17) for the gradient to reduce memory usage.

(4) Barycenter of Multiple Graphs

Given graphs $\{G^{(m)} \mid m = 1,\ldots,M\}$, the graph that minimizes the sum of dissimilarities between them is called the barycenter. If MFGW is employed as a dissimilarity measure, the barycenter graph $\bar{G}$ can be defined as $\bar{G} = \operatorname{argmin}_G \sum_{m=1}^{M} \mathrm{MFGW}(G, G^{(m)})$. Given the number of nodes in G, the node-attributes, edge-attributes, and edge weights can be optimized via the gradient descent methods using the gradient of MFGW. We can reveal common structures inherent in multiple graphs using the barycenter.

5. Experiments

In this section, we experimentally discuss the effectiveness of MFGW on representative applications. The graphs are assigned the one-hot vectors of discrete labels or normalized continuous values as the node- and edge-attributes. If both discrete labels and continuous values are available, the continuous values are used for the attributes. We have employed edge weight values for the first dimension of edge-embedding and $\beta = 0.5$ unless otherwise stated.

In all experiments, we set $\alpha = 0.1$, $\epsilon_{\text{target}} = 0.01$, $\epsilon_{\text{init}} = 1000$ and $p, q = 2$. The amount of noise used in the optimization varied depending on the costs as $\sigma^{(t)} = 0.00001 * \max_{ik} D^{(t)}_{ik}$. The number of updates T is set to 20, and the number of the generalized Sinkhorn iterations is set to 5,000. In the subgraph matching task, the masses are set to $a_i = a_j = 1/\max(N, \tilde{N})$ for all i, j. For the other tasks, the masses are set to $a_i = \frac{1}{N}$ and $\tilde{a}_j = \frac{1}{\tilde{N}}$ for all i, j. MFGW was implemented using PyTorch [30].

5.1 Supervised Classification

We compare MFGW kernel with the representative graph kernel methods; FGW kernel (FGW) [1], shortest path kernel (SP) [9], Weisfeiler-Lehman Subtree Kernel (WL) [12], Wasserstein Weisfeiler-Lehman Kernel (WWL) [16], and Deep Divergence Graph Kernels (DDGK) [20] in the performance of graph classification[†]. We implemented SP and WL with GraKeL [32], while FGW, WWL and DDGK implementations were beased on the official releases [1], [16], [20]. Their performance was evaluated by training SVMs [33], employing six datasets with edge-attributes, AIDS (2000) [34], Cuneiform (267) [35], MUTAG (188) [36], [37], PTC MR (344) [13], BZR MD (306), and COX2 MD (303) [37], [38], where the numbers in the parentheses indicates the numbers of graphs in the datasets, respectively. All datasets were obtained from [39]. The parameter of SVM (C) was optimized in the range $\{10^{-7}, 10^{-6}, \ldots, 10^{7}\}$ and the kernel parameter λ is optimized in the range $\{2^{-7}, 2^{-6}, \ldots, 2^{7}\}$ using 10-fold nested cross-validation (CV) [40] as in [1]. To mitigate the impact of dataset splitting during the CV, the same splitting was used for all methods. For the intra-graph similarities in MFGW and FGW, two settings were employed: one uses the edge weight values and the other uses the shortest path distances. Note that all edge weight values are binaries in all the datasets. Since both BZR MD and COX2 MD contain only complete graphs, we evaluated only the setting using the edge weight values. The SVMs were implemented using the scikit-learn [33]. The classification accuracy is shown in Table 1. MFGW achieved the best performance in three out of six datasets. Furthermore, MFGW outperformed FGW in nine of the ten settings.

By optimizing β for each dataset, MFGW can improve the accuracy, while MFGW with $\beta = 0.5$ already has a higher performance than existing methods as shown in Table 1. We can estimate the optimal β using the nested cross-validation as well as the other parameters if the number of graphs in the dataset is sufficient. Table 1 shows the accuracy obtained from MFGW with tuned β in range $\{0.0, 0.1, \ldots, 1.0\}$. As shown in the table, the performance of MFGW with tuned β is not better than that of MFGW with $\beta = 0.5$, except for PTC MR dataset. This is because the datasets used in this experiment had small numbers of graphs for tuning β. If the dataset contains a sufficient number of graphs, β can be tuned appropriately with the validation set similar to PTC MR dataset.

The prediction accuracy was also evaluated for datasets with no edge-attributes (BZR, COX2 [38], ENZYMES [41], [42], Synthetic [43], IMDB-B, and IMDB-M [44]). The accuracy is shown in Table 2. Even for graphs without edge-attributes, MFGW showed competitive performance with the existing graph kernels. Compared with FGW, MFGW achieved higher classification accuracy in nine of the twelve settings. These results indicate that our heuristics lead to better performance in the downstream task.

[†]For SP, WL, and DDGK, discrete node-labels are used for the node-attributes if both discrete labels and continuous values are available because these methods handle only discrete labels. We employed RBF kernel for DDGK.

Table 1 Classification accuracy in nested 10-fold CV

Method	Edge-attributed graph				Complete graph	
	AIDS	Cuneiform	MUTAG	PTC MR	BZR MD	COX2 MD
MFGW(W)	99.60 ± 0.37	**88.15**± 6.58	88.95 ± 7.24	58.86 ± 6.29	**72.26** ± 6.64	**68.71** ± 6.77
MFGW (SP)	99.65 ± 0.32	85.93 ± 8.73	88.42 ±8.42	54.00 ±5.49	-	-
FGW (W) [1]	99.55 ±0.35	55.19 ±9.14	84.21 ±7.81	58.29 ±5.88	70.97 ± 7.07	64.84 ± 10.14
FGW (SP) [1]	99.30 ± 0.60	57.04 ± 8.15	84.74 ± 7.61	56.86 ± 3.47	-	-
WL [12]	99.55 ± 0.27	71.85 ± 6.24	89.47 ± 4.71	59.14 ± 5.72	61.94 ± 6.42	57.42 ± 6.09
SP [9]	**99.75** ± 0.34	72.96 ± 5.51	88.42 ± 5.67	56.86 ± 10.26	70.32 ± 6.26	65.81 ± 7.80
WWL [16]	99.50 ± 0.55	59.63 ± 10.27	**90.53** ± 6.57	**60.29** ± 6.57	62.90 ± 6.09	48.71 ± 4.88
DDGK [20]	99.35 ± 0.63	15.56 ± 3.99	86.32 ± 6.74	56.57 ± 7.32	66.77 ± 7.22	48.06 ± 4.20
MFGW tuned β (W)	99.60 ± 0.49	81.85 ± 7.50	87.37± 7.51	60.29 ± 7.05	72.26 ± 8.19	66.13 ± 8.92
MFGW tuned β (SP)	99.60 ± 0.37	81.48 ± 7.94	88.42 ± 9.01	55.43 ± 5.60	-	-

Table 2 Classification accuracy in nested 10-fold CV (No edge-attributes)

Method	No edge-attributes				No attributes	
	BZR	COX2	ENZYMES	Synthetic	IMDB-B	IMDB-M
MFGW (W)	86.34 ± 4.78	79.36 ± 3.69	69.17 ± 6.29	94.33 ± 3.35	71.40 ± 3.58	47.73 ± 3.63
MFGW (SP)	85.85 ± 6.25	79.57 ± 5.14	66.50 ± 3.76	**96.33** ± 3.48	71.00 ± 5.25	47.73 ± 3.75
FGW (W) [1]	84.88 ± 4.85	78.09 ± 5.04	70.83 ± 6.34	49.33 ± 9.52	70.00 ± 4.17	47.93 ± 3.41
FGW (SP) [1]	84.15 ± 5.80	77.40 ± 4.17	70.67 ± 6.76	84.33 ± 6.84	66.10 ± 2.70	46.93 ± 4.20
WL [12]	**88.78** ± 4.39	79.15 ± 4.93	56.50 ± 4.44	42.67 ± 4.16	71.00 ± 3.61	50.67 ± 3.29
SP [9]	86.59 ± 4.79	**80.21** ± 3.44	44.33± 6.06	42.67 ± 4.16	71.50 ± 2.54	49.87 ± 4.00
WWL [16]	83.90 ± 5.69	75.53 ± 4.18	**77.50** ± 4.79	69.00 ± 8.57	**72.30** ± 2.53	**51.13** ± 3.14
DDGK [20]	81.46 ± 6.65	77.66 ± 5.23	24.5 ± 5.82	53.67 ± 6.23	63.80 ± 5.02	43.33 ± 4.06

5.2 Subgraph Matching

We demonstrate the effectiveness of our optimization heuristics in subgraph matching tasks. Since the solution of Eq. (3) corresponds subgraph matching if all masses are the same, we evaluate the solutions obtained by MFGW with the errors in the subgraph matching task. We evaluate MFGW, MFGW without heuristics, and FGW by using the six graph datasets having edge-attributes. All graphs in the datasets were used as targets, and a query is generated for each target. The query is a subgraph containing nodes within a 2-hop distance from a randomly selected node in the target.

The errors were evaluated by averaging the Frobenius norm of the difference between the ground truth (i.e., the optimal solution of Eq. (3)) and the correspondence matrices obtained by using MFGW and FGW. We calculated the ground truth using VF2 algorithm [23] implemented in NetworkX [45]. If multiple subgraphs in the target matched the query, the one having the smallest Frobenius norm with the correspondence matrix was taken as the ground truth for each method.

The errors are shown in Table 3. The table shows that our heuristics effectively reduce the errors. Since the errors correspond to the difference between the optimal solution of Eq. (3) and the obtained solution, this indicates that our heuristics are effective in obtaining solutions close to the optimal solution. In addition, FGW yields a larger error than MFGW including the "MFGW w/o edge-attributes" setting. This indicates that our heuristics are effective even if the edge-attributes are not available.

We then demonstrate the effectiveness of MFGW on the similar subgraph search task. We tested similar subgraph searches on BZR MD dataset by using MFGW and the state-of-the-art subgraph matching method, VF2 [23]. In the dataset, the graphs have continuous edge-attributes and discrete node-labels. Three subgraphs in the dataset was used as the queries. The subgraph similarity search can be performed by computing the MFGW between the query and each graph in the dataset. The queries and the obtained graphs (top-5) are shown in Fig. 3. For clarity, only edges corresponding to chemical bonds are shown in the figure, although BZR MD contains edge information regardless of the presense of chemical bonds. In Fig. 3, the values on the edges represent the edge-attributes and the node colors represent the node-labels. For simplicity, edge-attributes are shown only on the subgraphs that match the query. The red lines denotes queries and matched subgraphs. As shown in the figures, MFGW successfully retrieved the subgraph that exactly matches the query in addition to subgraphs with slightly different edge-attributes. The graphs obtained by VF2 (threshold=0) [23], which are the subgraphs that exactly match to the query, are also shown in Fig. 3. It is difficult to set an appropriate threshold for continuous attributes. If the threshold is too small, similar subgraphs cannot be obtained as shown in Fig. 3. Conversely, if the threshold is too large, too many subgraphs will be outputted. Unlike First [22] and VF2 [23], MFGW is effective to find similar subgraph matchings with continuous edge-attributes. This is because MFGW can directly evaluate the similarity without specifying any thresholds even if graphs have continuous edge-attributes.

5.3 Barycenter Graph

Finally, barycenters were obtained to demonstrate the effec-

Table 3 Subgraph matching error (×10)

Method	AIDS	Cuneiform	MUTAG	PTC MR	BZR MD	COX2 MD
MFGW	**0.268** ± 0.673	**0.000** ± 0.000	**0.570** ± 0.830	**0.416** ± 0.766	**0.000** ± 0.000	**0.019** ± 0.141
MFGW w/o noise	0.614 ± 0.827	0.600 ± 0.401	0.708 ± 0.771	0.933 ± 0.936	0.082 ± 0.207	0.594 ± 0.170
MFGW w/o deterministic annealing	0.797 ± 1.112	0.239 ± 0.591	0.986 ± 0.887	0.931 ± 1.098	1.450 ± 0.844	1.258 ± 0.704
MFGW w/o edge-attributes	0.651 ± 0.997	0.016 ± 0.187	0.756 ± 0.780	0.631 ± 0.942	2.818 ± 0.320	2.379 ± 0.165
FGW	1.216 ± 0.913	0.739 ± 0.461	1.080 ± 0.593	1.398 ± 0.932	2.006 ± 0.202	1.766 ± 0.078

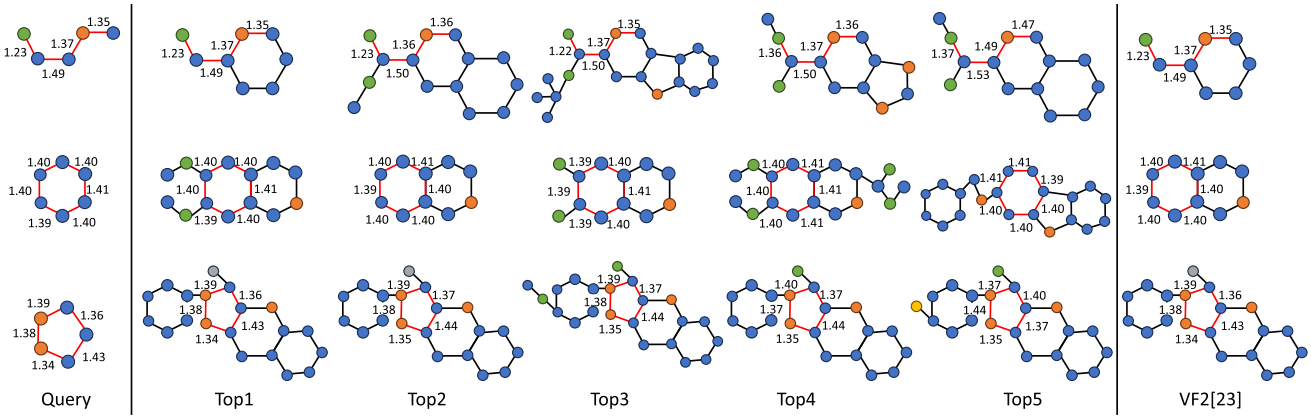
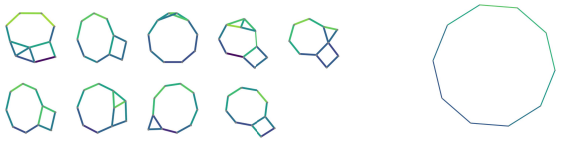


Fig. 3 Subgraph similarity search with MFGW (Top-5).



(a) Inputs (b) Barycenter

Fig. 4 Barycenter graph obtained by MFGW

tiveness of MFGW. Figure 4 (a) illustrates the input graphs generated by randomly adding edges to circle graphs with noisy edge-attributes. With the gradient descent, we optimized the edge-attributes and edge weight values of the randomly initialized barycenter graph having ten nodes; employing Adam (the learning rate is 0.1) [46]. The number of iterations in the gradient descent was set to 200. The graph having the lowest cost during the optimization was chosen for the barycenter. The obtained graph is displayed in Fig. 4 (b). As shown in Fig. 4 (b), the barycenter graph could be a denoised graph. Note that only the node-attributes and edge weights can be optimized with FGW, while the edge-attributes can also be optimized with MFGW.

6. Discussion

(1) Importance of Edge-Attribute

MFGW balances intra-graph similarity and edge-attributes with the parameter β. The importance of edge-attributes and intra-graph similarity (i.e., the optimal β) varies depending on the dataset. For example, in Sect. 5.1, the accuracy of BZR MD dropped from 72.26 to 66.45 (−5.81 points) by varying β from 0.5 to 1.0 in MFGW (W). This indicates that edge-attributes are more important for the BZR MD dataset than intra-graph similarity. Conversely, for MUTAG dataset, the accuracy increased from 88.95 to 90.00 (+1.05 points) by varying β from 0.5 to 1.0. This indicates that intra-graph similarity is more important than edge-attributes for the MUTAG dataset. If the dataset has enough data, the parameter β can be tuned with the validation set. Even if the tuning is difficult, for example due to a small number of data in the dataset, MFGW with $\beta = 0.5$ performs better than the existing kernels, as shown in Sect. 5.1.

(2) Distance between Edge-Labels

In the experiments in Sect. 5, the distance between one-hot encoded vectors of the labels are used for the distance between edge-labels. If some domain knowledge is available (e.g., similarity between interatomic bonds), it can be used to improve the quality of distances between edge-labels resulting in high performance of subsequent tasks. This is especially effective for datasets with large variations in the distances between edge-labels. Furthermore, embeddings of edge-labels can be optimized using a loss function of the subsequent task (e.g., classification) using the gradient of the MFGW, similar to Sect. 5.3. However, this paper focuses on MFGW, which is the distance between edge-attributed graphs, and the customization of the distance between edge-labels is our future work.

7. Conclusion

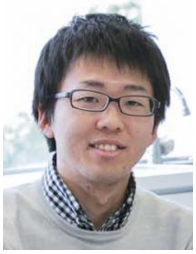
In this paper, we proposed MFGW that is an OT dissimilarity between graphs with edge-attributes. MFGW is an extension of FGW, and can directly handle edge-attributes. We also proved that MFGW is pseudometric on edge-attributed graphs, allowing the use of existing machine learning meth-

ods. Heuristic techniques were also proposed for the non-convex quadratic optimization required for MFGW. We experimentally demonstrated that MFGW outperforms the existing methods. An issue with MFGW is that the parameter α and the attributes, $\mathbf{x}_i$ and $\mathbf{e}_{ij}$, should be properly designed by the users. Optimizing these values for each task such as classification, is the continuation of our work in this domain.

References

[1] 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.
[2] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," J. machine learning research, vol.9, no.11, 2008.
[3] H. Xu, D. Luo, H. Zha, and L.C. Duke, "Gromov-wasserstein learning for graph matching and node embedding," Int. Conf. machine learning, pp.6932–6941, PMLR, 2019.
[4] H. Xu, D. Luo, and L. Carin, "Scalable gromov-wasserstein learning for graph partitioning and matching," Advances in neural information processing systems, vol.32, 2019.
[5] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," Advances in neural information processing systems, vol.26, 2013.
[6] J. Solomon, G. Peyré, V.G. Kim, and S. Sra, "Entropic metric alignment for correspondence problems," ACM Trans. Graphics (ToG), vol.35, no.4, pp.1–13, 2016.
[7] S. Ferradans, N. Papadakis, G. Peyré, and J.F. Aujol, "Regularized discrete optimal transport," SIAM J. Imaging Sciences, vol.7, no.3, pp.1853–1882, 2014.
[8] R. Flamary, N. Courty, A. Rakotomamonjy, and D. Tuia, "Optimal transport with laplacian regularization," NIPS 2014, Workshop on Optimal Transport and Machine Learning, 2014.
[9] K.M. Borgwardt and H. Kriegel, "Shortest-path kernels on graphs," Fifth IEEE Int. Conf. data mining (ICDM'05), pp.74–81, IEEE, 2005.
[10] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," Proc. 20th Int. Conf. machine learning (ICML-03), pp.321–328, 2003.
[11] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," Learning theory and kernel machines, pp.129–143, Springer, 2003.
[12] N. Shervashidze, P. Schweitzer, E.J. Van Leeuwen, K. Mehlhorn, and K.M. Borgwardt, "Weisfeiler-lehman graph kernels.," J. Machine Learning Research, vol.12, no.9, 2011.
[13] C. Helma, R.D. King, S. Kramer, and A. Srinivasan, "The predictive toxicology challenge 2000–2001," Bioinformatics, vol.17, no.1, pp.107–108, 2001.
[14] G. Peyré and M. Cuturi, "Computational optimal transport: With applications to data science," Foundations and Trends® in Machine Learning, vol.11, no.5-6, pp.355–607, 2019.
[15] F. Mémoli, "Gromov–wasserstein distances and the metric approach to object matching," Foundations of computational mathematics, vol.11, no.4, pp.417–487, 2011.
[16] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt, "Wasserstein weisfeiler-lehman graph kernels," Advances in Neural Information Processing Systems, vol.32, 2019.
[17] S. Kolouri, N. Naderializadeh, G.K. Rohde, and H. Hoffmann, "Wasserstein embedding for graph learning," Int. Conf. Learning Representations, 2020.
[18] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," arXiv preprint arXiv:1810.00826, 2018.
[19] W. Liu, H. Qian, C. Zhang, J. Xie, Z. Shen, and N. Zheng, "From one to all: Learning to match heterogeneous and partially overlapped graphs," Proc. AAAI Conf. Artificial Intelligence, vol.36, no.4, pp.4109–4119, Jun. 2022.
[20] R. Al-Rfou, B. Perozzi, and D. Zelle, "Ddgk: Learning graph representations for deep divergence graph kernels," The World Wide Web Conference, pp.37–48, 2019.
[21] S. Zhang and H. Tong, "Final: Fast attributed network alignment," Proc. 22nd ACM SIGKDD Int. Conf. knowledge discovery and data mining, pp.1345–1354, 2016.
[22] B. Du, S. Zhang, N. Cao, and H. Tong, "First: Fast interactive attributed subgraph matching," Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, KDD '17, New York, NY, USA, p.1447–1456, Association for Computing Machinery, 2017.
[23] L.P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," IEEE Trans. pattern analysis and machine intelligence, vol.26, no.10, pp.1367–1372, 2004.
[24] J. Li, J. Tang, L. Kong, H. Liu, J. Li, A.M.C. So, and J. Blanchet, "Fast and provably convergent algorithms for gromov-wasserstein in graph learning," arXiv preprint arXiv:2205.08115, 2022.
[25] Y. Xie, X. Wang, R. Wang, and H. Zha, "A fast proximal point method for computing exact wasserstein distance," Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, ed. R.P. Adams and V. Gogate, Proceedings of Machine Learning Research, vol.115, pp.433–453, PMLR, 22–25 Jul 2020.
[26] G. Peyré, M. Cuturi, and J. Solomon, "Gromov-wasserstein averaging of kernel and distance matrices," Int. Conf. machine learning, pp.2664–2672, PMLR, 2016.
[27] S. Chowdhury, D. Miller, and T. Needham, "Quantized gromov-wasserstein," Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21, pp.811–827, Springer, 2021.
[28] G. Shakhnarovich, T. Darrell, and P. Indyk, "Nearest-neighbor methods in learning and vision: theory and practice," Neural Information Processing, The MIT press, 2006.
[29] V. Titouan, N. Courty, R. Tavenard, and R. Flamary, "Optimal transport for structured data with application on graphs," Int. Conf. Machine Learning, pp.6275–6284, PMLR, 2019.
[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems, vol.32, 2019.
[31] G. Luise, A. Rudi, M. Pontil, and C. Ciliberto, "Differential properties of sinkhorn approximation for learning with wasserstein distance," Advances in Neural Information Processing Systems, vol.31, 2018.
[32] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis, "Grakel: A graph kernel library in python," J. Machine Learning Research, vol.21, no.54, pp.1–5, 2020.
[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," J. Machine Learning Research, vol.12, pp.2825–2830, 2011.
[34] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," Structural, Syntactic, and Statistical Pattern Recognition, ed. N. da Vitoria Lobo, T. Kasparis, F. Roli, J.T. Kwok, M. Georgiopoulos, G.C. Anagnostopoulos, and M. Loog, Berlin, Heidelberg, pp.287–297, Springer Berlin Heidelberg, 2008.
[35] N.M. Kriege, M. Fey, D. Fisseler, P. Mutzel, and F. Weichert, "Recognizing cuneiform signs using graph based methods," Int. Workshop on Cost-Sensitive Learning, pp.31–44, PMLR, 2018.
[36] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity," J. medicinal

chemistry, vol.34, no.2, pp.786–797, 1991.
[37] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," Proceedings of the 29th Int. Conf. Machine Learning, ICML'12, Madison, WI, USA, p.291–298, Omnipress, 2012.
[38] J.J. Sutherland, L.A. O'Brien, and D.F. Weaver, "Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships," J. chemical information and computer sciences, vol.43, no.6, pp.1906–1915, 2003.
[39] K. Kersting, N.M. Kriege, C. Morris, P. Mutzel, and M. Neumann, "Benchmark data sets for graph kernels," 2016.
[40] G.C. Cawley and N.L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," The J. Machine Learning Research, vol.11, pp.2079–2107, 2010.
[41] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg, "Brenda, the enzyme database: updates and major new developments," Nucleic acids research, vol.32, no.suppl_1, pp.D431–D433, 2004.
[42] K.M. Borgwardt, C.S. Ong, S. Schönauer, S.V.N. Vishwanathan, A.J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," Bioinformatics, vol.21, no.suppl_1, pp.i47–i56, 2005.
[43] A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. Borgwardt, "Scalable kernels for graphs with continuous attributes," Advances in neural information processing systems, vol.26, 2013.
[44] P. Yanardag and S.V.N. Vishwanathan, "Deep graph kernels," Proc. 21th ACM SIGKDD Int. Conf. knowledge discovery and data mining, pp.1365–1374, 2015.
[45] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
[46] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.



Keisuke Kawano received the B.E. and M.E. degrees in mechanical engineering from Osaka University, Osaka, Japan, in 2013 and 2015, respectively. Since 2015, he has been a researcher in Toyota Central R&D Labs., Inc., Aichi, Japan. His current research interests include machine learning, artificial intelligence, and their applications.

Satoshi Koide received the B.S. and M.S. degrees in mathematics from Osaka University, Osaka, Japan, in 2004 and 2006, respectively. He received the Ph.D. degree in informatics from Nagoya University, Aichi, Japan, in 2020. Since 2006, he has been a researcher in Toyota Central R&D Labs., Inc., Aichi, Japan. His research interests include database systems, spatial data structures and algorithms, machine learning methodologies, mathematical optimization, and their real world applications.

Hiroaki Shiokawa is an Associate Professor at University of Tsukuba. He received B.S., M.E., and Ph.D in engineering from University of Tsukuba in 2009, 2011 and 2015, respectively. From 2011 to 2015, he was a research scientist at Nippon Telegraph and Telephone Corporation, and he joined Center for Computational Sciences at University of Tsukuba in Nov. 2015. His current research interests include database systems, data engineering, data mining, and graph data management.

Toshiyuki Amagasa received B.E., M.E., and Ph.D from the Department of Computer Science, Gunma University in 1994, 1996, and 1999, respectively. He is currently a full professor at the Center for Computational Sciences and the Center for Artificial Intelligence Research, University of Tsukuba. His research interests cover database systems, data mining, and database application in scientific domains. He is a senior member of IPSJ, IEICE, and IEEE, a board member of DBSJ, and a member of ACM.