WatE: A Wasserstein t-distributed Embedding Method for Information-enriched Graph Visualization

Minjie Cheng[1], **Dixin Luo**[2], **Hongteng Xu**[1,3*]

[1]Gaoling School of Artifcial Intelligence, Renmin University of China
[2]School of Computer Science and Technology, Beijing Institute of Technology
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
hongtengxu@ruc.edu.cn

Abstract

As a fundamental problem of graph analysis, graph visualization aims to embed a set of graphs in a low-dimensional (e.g., 2D) space and provide insights into their distribution and clustering structure. Focusing on this problem, we propose a novel Wasserstein t-distributed embedding (WatE) method, leading to an information-enriched graph visualization paradigm. Our method learns a graph neural network to represent each graph as the mean and covariance of its node embedding distribution. Accordingly, our method can visualize each graph as an ellipse (determined by the mean and the covariance) rather than a single point. The positions of different ellipses reveal the relations among different graphs as traditional visualization methods do, while the size and shape of an ellipse preserve the node-level structural information of the corresponding graph. We propose a regularized t-distributed stochastic neighbor embedding (Rt-SNE) framework to learn the visualization model, deriving a Wasserstein distance-based Student's t-distribution of graph pairs and fitting the distribution to the data distribution under regularization. Both subjective and objective evaluations demonstrate that WatE achieves encouraging performance in various graph visualization and clustering tasks.

Code — https://github.com/minjiecheng/WatE

Introduction

In many application scenarios, we often need to explore the distribution of graphs and visualize the graphs in a low-dimensional space, which may provide valuable insights for downstream tasks. In drug development, each molecule can be represented as a graph whose nodes are atoms and edges are chemical bonds. Given multiple molecules, capturing their clustering structure through visualization is significant for toxicological analysis and virtual screening. For social networks, each community corresponds to a subgraph of users. It is beneficial for social network analysis to visualize different communities based on their topological similarity.

Solving the above visualization problems requires us to develop a graph embedding model that represents graphs in a latent space. Given a graph, most existing methods (Xu et al. 2018; Sun et al. 2019; You et al. 2020; Suresh et al. 2021) aggregate its node embeddings (via various readout or global pooling operations) as its representation. Combining these embedding methods with traditional visualization methods, e.g., t-SNE (Van der Maaten and Hinton 2008) often leads to encouraging graph visualization results, which can capture graph-level clustering structures. However, this strategy loses the node-level structural information due to the aggregation of node embeddings — each graph is embedded as a single point in the latent space. Thus, the distribution of its node embeddings is ignored. On the contrary, although some attempts have been made to represent a graph via the set of node embeddings directly (Grover and Leskovec 2016; Kipf and Welling 2017; Petric Maretic et al. 2019), such set-level representations significantly increase the difficulties in graph visualization. In particular, when representing each graph as a set of node embeddings, we have to compute the distance/similarity between two arbitrary graphs by comparing their node embedding sets with high computational complexity. Additionally, when visualizing multiple graphs, we need to consider the hierarchical structure of their node embeddings, which requires us to ensure the visualization reflects both the graph-level clustering structure and the node-level distribution within each graph.

To overcome the above challenges, we propose a novel Wasserstein t-distributed Embedding (WatE) method in this study, which provides a new learning paradigm for graph visualization. As illustrated in Figure 1, our WatE method

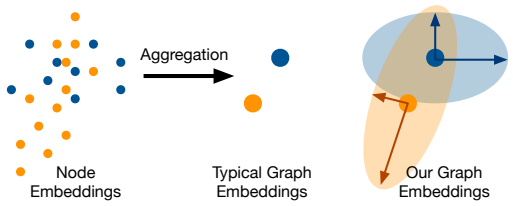


Figure 1: llustrations of typical graph embeddings and our approach. For our graph embeddings, the ellipses are determined by the corresponding mean vectors and covariance matrices related to the graph, while the arrows represent the principal components of these covariance matrices, incorporating more graph information.

*Corresponding author.

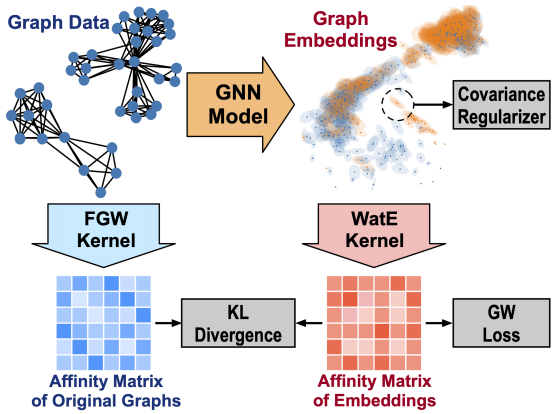


Figure 2: The learning scheme of our WatE. The three objectives in gray frames are used to learn a GNN-based embedding model. Our approach is an inductive learning paradigm specifically designed for graph visualization.

neither embeds a graph as a single point nor represents it as a set of node embeddings. Instead, given a graph, our method derives its node embeddings through a graph neural network (GNN) and visualizes it as an ellipse in 2D space, whose position and shape are determined by the mean and covariance of the node embeddings, respectively. This visualization strategy combines the advantage of the classic graph embedding methods (Xu et al. 2018; Sun et al. 2019; You et al. 2020) and that of the set-level embedding methods (Bachmann, Hennig, and Kobak 2022). In particular, given such distribution-based graph embeddings, the positions of the corresponding ellipses indicate the relations among the graphs as those classic visualization methods do, and the shape of each ellipse shows the node-level structural information of a graph.

Leveraging optimal transport techniques, we apply a regularized t-distributed stochastic neighbor embedding (Rt-SNE) framework to learn the proposed graph visualization model, as illustrated in Figure 2. In particular, for two arbitrary graphs, we leverage the Wasserstein distance between their node embedding distributions (Villani 2009) to measure their discrepancy, which has a closed-form solution and can be computed in a differentiable way via Newton-Schulz iterative algorithm (Muzellec and Cuturi 2018). Based on the Wasserstein distance, we can obtain the Student's t-distribution of graph pairs and fit it to the data distribution derived by the Fused Gromov-Wasserstein (FGW) distance (Titouan et al. 2019) between raw graphs, which leads to the KL-divergence loss used in t-SNE (Van der Maaten and Hinton 2008). To suppress the risk of over-fitting, when minimizing the KL-divergence loss we further i) consider the Gromov-Wasserstein-based generalized spectral clustering loss (Xu, Luo, and Carin 2019; Chowdhury and Needham 2021) to enhance the clustering structure of the graph embeddings, and ii) regularize the energy of the covariance by penalizing their discrepancies to the identity matrix. We test the WatE model and its Rt-SNE learning method in various graph visualization tasks. For the learned model, we analyze its robustness, generalization power, quantitative comparisons on clustering, and complexity in depth, demonstrating the rationality of the learning method and the effectiveness of the graph visualization model. Both subjective visualization and objective evaluation show that WatE provides a new and promising solution to achieve information-enriched graph visualization.

Related Work

Graph Embedding and Visualization

Graph embedding typically has two paradigms, and the first one is the kernel-based approach. The typical graph kernels include the Graphlet kernel (GK) (Shervashidze et al. 2009), random walk kernel (RWK) (Gärtner, Flach, and Wrobel 2003), shortest path kernel (SPK) (Borgwardt and Kriegel 2005), multi-scale Laplacian kernel (Kondor and Pan 2016), the Weisfeiler-Lehman kernel (WLK) (Shervashidze et al. 2011), and so on. Applying spectral clustering (Ng, Jordan, and Weiss 2001), we derive graph embeddings based on these kernels. The above kernel-based strategies are transductive and own quadratic computational complexity, whose scalability is questionable. To solve these problems, graph neural networks (GNNs) have been proposed to obtain graph embeddings explicitly and inductively, e.g., GCN (Kipf and Welling 2017), GIN (Xu et al. 2018), and unsupervised InfoGraph (Sun et al. 2019), GraphCL (You et al. 2020), and ADGCL (Suresh et al. 2021).

Given graph embeddings, we can visualize their distribution via various dimensionality reduction methods, such as principal component analysis (PCA) (Abdi and Williams 2010), ISOMAP (Tenenbaum, Silva, and Langford 2000), t-SNE (Van der Maaten and Hinton 2008), and so on. These visualization methods simplify graphs into single points, losing crucial node-level details. Additionally, they are transductive, requiring model retraining for new graph embeddings. Moreover, without additional mechanisms, node embeddings from different graphs may heavily overlap in low-dimensional spaces, impairing the visualization of clustering structures.

Optimal Transport for Graph Modeling

Recently, optimal transport theory (Villani 2009) has garnered attention in machine learning and data mining, leading to various methods such as Wasserstein GAN (WGAN) (Arjovsky, Chintala, and Bottou 2017) and Wasserstein antoencoder (Tolstikhin et al. 2018). These methods use Wasserstein distance to measure the discrepancy between the data distribution and the model distribution. In graph analysis, Gromov-Wasserstein (GW) distance (Mémoli 2011) and its variant fused Gromov-Wasserstein (FGW) distance (Titouan et al. 2019) are proposed, offering pseudo-metrics for graphs. Various algorithms, including Sinkhorn scaling (Cuturi 2013), the Bregman ADMM algorithm (Wang and Banerjee 2014), the proximal point method (Xie et al. 2020), and conditional gradient (Titouan et al. 2019), can efficiently compute these optimal transport-based distances.

Optimal transport-based distances have inspired effective graph modeling methods for embedding, clustering, and vi-

sualization. The work in (Titouan et al. 2019) considers the fused Gromov-Wasserstein distance between graphs and constructs a pseudo-kernel for graph spectral clustering. For unsupervised graph representation, the Gromov-Wasserstein factorization (GWF) model in (Xu et al. 2023) learns a set of graph factors and represents observed graphs via the weighted GW barycenters (Peyré, Cuturi, and Solomon 2016) of the graph factors. Following this strategy, a graph dictionary learning (GDL) method is proposed in (Vincent-Cuaz et al. 2021), which learns a linear graph factorization model under the GW distance metric. Recently, a Wasserstein t-SNE method (Bachmann, Hennig, and Kobak 2022) is proposed to visualize the data with hierarchical structures (e.g., sets of points). It extends the traditional t-SNE method using the Wasserstein distance to capture the discrepancy between different sets.

Method

Preliminaries. For high-dimensional data points, the t-distributed stochastic neighbor embedding (t-SNE) method in (Van der Maaten and Hinton 2008) provides an effective solution to visualize the data and explore their clustering structure in a low-dimensional space. Given N data points in the D-dimensional space, denoted as $\boldsymbol{X} = [\boldsymbol{x}_n] \in \mathbb{R}^{D\times N}$, the t-SNE method learns the low-dimensional embeddings via fitting the distribution of the embedding pairs to the distribution of the raw data pairs, i.e.,

$$\min_{\boldsymbol{Y}} \mathrm{KL}(\boldsymbol{P}(\boldsymbol{X})\|\boldsymbol{Q}(\boldsymbol{Y})) = \min_{\boldsymbol{Y}} \sum\nolimits_{n\neq n'} p_{nn'} \log \frac{p_{nn'}}{q_{nn'}}, \quad (1)$$

where $\boldsymbol{Y} = [\boldsymbol{y}_n] \in \mathbb{R}^{d\times N}$ represents the target low-dimensional embeddings and $d \ll D$. The affinity matrices $\boldsymbol{P}(\boldsymbol{X}) = [p_{nn'}]$ and $\boldsymbol{Q}(\boldsymbol{Y}) = [q_{nn'}]$ represent the distribution of the data and that of the embeddings, respectively, which are defined as follows:

$$\begin{aligned} p_{nn'} &= \frac{1}{2N}(p_{n'|n} + p_{n|n'}),\ \forall\, n, n' \in \{1, ..., N\},\ \text{with} \\ p_{n'|n} &= \begin{cases} \frac{\exp(-\|\boldsymbol{x}_n - \boldsymbol{x}_{n'}\|^2/(2\sigma_n^2))}{\sum_{k\neq n}\exp(-\|\boldsymbol{x}_n - \boldsymbol{x}_k\|^2/(2\sigma_n^2))} & n \neq n', \\ 0 & n = n', \end{cases} \\ q_{nn'} &= \begin{cases} \frac{(1+\|\boldsymbol{y}_n - \boldsymbol{y}_{n'}\|^2)^{-1}}{\sum_k \sum_{k\neq l}(1+\|\boldsymbol{y}_k - \boldsymbol{y}_l\|^2)^{-1}} & n \neq n', \\ 0 & n = n'. \end{cases} \end{aligned} \quad (2)$$

Applying stochastic gradient descent, the t-SNE method solves (1) and visualizes the data by the learned embeddings. Although the t-SNE method has been widely used in many visualization tasks, when the data are a set of graphs rather than vectors, it becomes inapplicable because the Euclidean distance used in (2) is undefined for graph-structured data. Additionally, the t-SNE method is transductive, in which the target embeddings are non-parametric. When new data comes, we have to retrain the model.

To overcome the challenges, we would like to extend the t-SNE framework, developing a new unsupervised and inductive graph visualization paradigm. In particular, we design a graph neural network-based embedding model and propose a regularized t-distributed stochastic neighbor embedding (Rt-SNE) framework to learn the model with the help of the computational optimal transport techniques (Titouan et al. 2019; Mémoli 2011; Rippl, Munk, and Sturm 2016). The graph embeddings derived by the proposed model should i) reflect the graph-level clustering structure of the corresponding graphs and ii) preserve the node-level structural information for each graph, leading to informative graph visualization results.

FGW-based Affinity Matrix of Graphs. Suppose that we have N graphs, denoted as $\mathcal{G} = \{G_n\}_{n=1}^N$. For a graph with M nodes, we can represent it as a *measurement graph* (Chowdhury and Mémoli 2019) i.e., $G := G(\boldsymbol{A}, \boldsymbol{V}, \boldsymbol{p})$, which consists of an adjacency matrix $\boldsymbol{A} = [a_{ij}] \in \{0,1\}^{M\times M}$, node feature matrix $\boldsymbol{V} \in \mathbb{R}^{M\times D}$, and a node distribution $\boldsymbol{p} \in \Delta^{M-1}$. The i-th row of $\boldsymbol{V}$, denoted as $\boldsymbol{v}_i$, is the feature of the node i. Δ^{M-1} denotes the $(M-1)$-Simplex. Here, the element $a_{ij} = 1$ means an edge exists between the node i and the node j. Following the work in (Xu, Luo, and Carin 2019), the node distribution is defined as the normalized node degrees, i.e., $\boldsymbol{p} = \frac{1}{\|\boldsymbol{A}\mathbf{1}_M\|_1}\boldsymbol{A}\mathbf{1}_M$, which reflects the significance of the nodes in the graph.

Our WatE method first constructs an affinity matrix (i.e., the $\boldsymbol{P}$ in (2)) to capture the pairwise similarity of graphs. The affinity matrix is defined based on fused Gromov-Wasserstein (FGW) distance (Titouan et al. 2019):

Definition 1 (Fused Gromov-Wasserstein distance)**.** *Let $G_1 = G(\boldsymbol{A}_1, \boldsymbol{V}_1, \boldsymbol{p}_1)$ and $G_2 = G(\boldsymbol{A}_2, \boldsymbol{V}_2, \boldsymbol{p}_2)$ be two attributed graphs. The fused Gromov-Wasserstein distance $d_{fgw}(G_1, G_2)$ is defined as*

$$\begin{aligned} &\min_{\boldsymbol{T}\in\Pi(\boldsymbol{p}_1,\boldsymbol{p}_2)} \underbrace{\sum_{i,j} d_V(\boldsymbol{v}_i^1, \boldsymbol{v}_j^2) t_{ij}}_{\textit{Wasserstein term}} + \underbrace{\sum_{i,j,i',j'} d_A(a_{ii'}^1, a_{jj'}^2) t_{ij} t_{i'j'}}_{\textit{Gromov-Wasserstein term}} \\ &= \min\nolimits_{\boldsymbol{T}\in\Pi(\boldsymbol{p}_1,\boldsymbol{p}_2)} \mathbb{E}_{(i,j)\sim\boldsymbol{T}}[d_V] + \mathbb{E}_{(i,j,i',j')\sim\boldsymbol{T}\times\boldsymbol{T}}[d_A], \end{aligned}$$

where $d_V(\boldsymbol{v}_i^1, \boldsymbol{v}_j^2)$ measures the discrepancy between the node features of the two graphs, $d_A(a_{ii'}^1, a_{jj'}^2)$ measures the discrepancy between the edges of the two graphs, and $\boldsymbol{T} = [t_{ij}]$ is the joint distribution of the nodes of the two graphs, and $\Pi(\boldsymbol{p}_1, \boldsymbol{p}_2) = \{\boldsymbol{T} \geq \mathbf{0} | \boldsymbol{T}\mathbf{1} = \boldsymbol{p}_1, \boldsymbol{T}^T\mathbf{1} = \boldsymbol{p}_2\}$ represents the set of the distributions of the node pairs that take $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ as their marginals.

The optimal joint distribution corresponding to the FGW distance, denoted as $\boldsymbol{T}^*$, is called the optimal transport between the two graphs. Its element t_{ij}^* indicates the coherency probability of the node i in G_1 and the node j in G_2.

Based on the FGW distance, we construct the affinity matrix $\boldsymbol{P}(\mathcal{G}) = [p_{nn'}] \in \mathbb{R}^{N\times N}$ for the graph dataset $\mathcal{G}$. Similar to (2), the $p_{nn'}$ is derived as follows:

$$\begin{aligned} p_{nn'} &= \frac{1}{2N}(p_{n|n'} + p_{n'|n}), \\ p_{n'|n} &= \begin{cases} \frac{\exp(-d_{\text{fgw}}^2(G_n, G_{n'})/(2\sigma_n^2))}{\sum_{k\neq n}\exp(-d_{\text{fgw}}^2(G_n, G_k)/(2\sigma_n^2))} & n \neq n', \\ 0 & n = n'. \end{cases} \end{aligned} \quad (3)$$

Here, each graph G_n is associated with a specific Gaussian kernel, whose bandwidth is σ_n. Following the t-SNE method (Van der Maaten and Hinton 2008), we determine

this bandwidth by the bisection search, making the entropy of the conditional distribution $\{p_{n'|n}\}_{n'=1}^{N}$ equal to a predefined perplexity. If the neighborhood of a graph G is dense, the corresponding σ will be small.

Wasserstein t-distribution of Embeddings. The affinity matrix $\boldsymbol{P}(\mathcal{G})$ guides the learning of the proposed graph embeddings. In particular, it captures the pairwise similarity of the graphs and thus reflects their clustering structure. The proposed graph embeddings should inherit the clustering structure in a low-dimensional space, whose affinity matrix should be close to $\boldsymbol{P}(\mathcal{G})$.

Our WatE method proposes a distribution-based graph embedding model and constructs the affinity matrix of the embeddings based on a Wasserstein-based t-distribution. Denote the proposed graph embedding model as f_θ, where θ represents the model parameters. Given a graph G with M nodes, this model derives the graph embedding as the mean and covariance of node embeddings, i.e.,

$$\begin{aligned} &\boldsymbol{X} = f_\theta(G) \in \mathbb{R}^{M\times K}, \quad \hat{\boldsymbol{\mu}} = \frac{1}{M}\boldsymbol{X}^T\mathbf{1}_M \in \mathbb{R}^K, \\ &\widehat{\boldsymbol{\Sigma}} = \frac{1}{M-1}(\boldsymbol{X}^T - \hat{\boldsymbol{\mu}}\mathbf{1}_M^T)(\boldsymbol{X} - \mathbf{1}_M\hat{\boldsymbol{\mu}}^T) \in \mathbb{R}^{K\times K}. \end{aligned} \tag{4}$$

Here, our model first takes the graph G as its input and outputs M K-dimensional node embeddings denoted as $\boldsymbol{X}$. Then, we treat the node embeddings as the samples of a latent distribution defined on the K-dimensional space. The $\hat{\boldsymbol{\mu}} \in \mathbb{R}^K$ and the $\widehat{\boldsymbol{\Sigma}} \in \mathbb{R}^{K\times K}$ in (4) are unbiased estimations of mean and covariance matrix, respectively.

Our WatE method takes the tuple $(\hat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ as the proposed graph embedding, which captures the statistics of the node embeddings' distribution. When $K = 2$, such distribution-based graph embeddings correspond to ellipses in 2D space, as illustrated in Figure 1. The positions of different ellipses are determined by the $\hat{\boldsymbol{\mu}}$'s, and the size and shape of each ellipse are determined by the $\widehat{\boldsymbol{\Sigma}}$ (through the singular value decomposition).[1] It is beneficial to visualize graphs through the distribution-based graph embeddings because the $\hat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ represent the graph G in different viewpoints. On the one hand, $\hat{\boldsymbol{\mu}}$ is the aggregation of the node embeddings. The positions of different $\hat{\boldsymbol{\mu}}$'s reveal the relationships among the corresponding graphs in the latent space, which captures the graph-level clustering structure. On the other hand, $\widehat{\boldsymbol{\Sigma}}$ reflects more node-level structural information within the graph, e.g., the distribution and concentricity of node embeddings. Leveraging $\hat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ jointly makes our WatE method have more discriminative power — when two graphs have similar $\hat{\boldsymbol{\mu}}$'s, the covariance matrices may help to distinguish them from each other. Given the graph embeddings, i.e., the $\hat{\boldsymbol{\mu}}$'s and $\widehat{\boldsymbol{\Sigma}}$'s, we measure the discrepancy between different embeddings via the Wasserstein distance (Villani 2009) between the corresponding latent distributions.

Definition 2 (Wasserstein distance)**.** *Let p and q be two probability measures on a compact metric space $(\mathcal{X}, d_X)$, where $\mathcal{X}$ denotes the space and d_X denotes the metric in the space. The Wasserstein distance between p and q is*

$$\begin{aligned} d_w(p,q) &= \inf_{\pi\in\Pi(p,q)}\big(\textstyle\int_{\mathcal{X}^2} d_X(x,y)^2 \mathrm{d}\pi(x,y)\mathrm{d}x\mathrm{d}y\big)^{\frac{1}{2}} \\ &= \inf_{\pi\in\Pi(p,q)} \mathbb{E}^{\frac{1}{2}}_{(x,y)\sim\pi}[d_X^2(x,y)], \end{aligned} \tag{5}$$

where $d_X(x,y)$ captures the distance between two arbitrary samples, and $\Pi(p,q)$ is the set of all probability measures on $\mathcal{X}\times\mathcal{X}$ with p and q as marginals, i.e., $\Pi(p,q) = \{\pi \geq 0 | \int_{\mathcal{X}} \pi(x,y)\mathrm{d}x = q(y),\ \int_{\mathcal{X}} \pi(x,y)\mathrm{d}y = p(x)\}$.

The Wasserstein distance finds the optimal probability measure in $\Pi(p,q)$ to minimize the expectation of $d_X(x,y)$. The optimal probability measure, i.e., $\pi^* = \arg\inf_{\pi\in\Pi(p,q)} \mathbb{E}_{(x,y)\sim\pi}[d_X(x,y)]$, works as the optimal transport plan between p and q. We use the Wasserstein distance in this study because it applies to the probability measures with non-overlapped support sets, which outperforms other metrics like KL-divergence and Jensen-Shannon divergence in such challenging scenarios (Arjovsky, Chintala, and Bottou 2017).

When p and q are Gaussian distributions, we can avoid solving (5) and derive the Wasserstein distance in a closed form (Rippl, Munk, and Sturm 2016):

Definition 3. *Let $p = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ and $q = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ be two K-dimensional Gaussian distributions, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and the covariance matrix, respectively. The Wasserstein distance between p and q, i.e., $d_w(p,q)$, is*

$$(\|\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\|_2^2 + tr(\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q - 2(\boldsymbol{\Sigma}_p^{\frac{1}{2}}\boldsymbol{\Sigma}_q\boldsymbol{\Sigma}_p^{\frac{1}{2}})^{\frac{1}{2}}))^{\frac{1}{2}}.$$

To simplify the computation, we assume that the node embeddings $\boldsymbol{X}$ in (4) obey a multivariate Gaussian distribution, i.e., $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. As a result, the $\hat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ derived by our graph embedding model are unbiased estimations of the distribution's parameters. Plugging the $\hat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\Sigma}}$ in (4) into Definition 3, we obtain a sample-based Wasserstein distance between the proposed graph embeddings.

Given the Wasserstein distances, we can construct the affinity matrix for the proposed graph embeddings by computing a Wasserstein-based t-distribution. In particular, the affinity matrix of the graph embeddings is a function of the model parameters, denoted as $\boldsymbol{Q}(\theta) = [q_{nn'}(\theta)]$, whose element $q_{nn'}(\theta)$ is defined as follows:

$$q_{nn'} = \begin{cases} \frac{(1+d_w^2(\mathcal{N}(\hat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n), \mathcal{N}(\hat{\boldsymbol{\mu}}_{n'}, \widehat{\boldsymbol{\Sigma}}_{n'})))^{-1}}{\sum_{k\neq l}(1+d_w^2(\mathcal{N}(\hat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k), \mathcal{N}(\hat{\boldsymbol{\mu}}_l, \widehat{\boldsymbol{\Sigma}}_l)))^{-1}} & n \neq n', \\ 0 & n = n', \end{cases} \tag{6}$$

where $d_w^2(\mathcal{N}(\hat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n), \mathcal{N}(\hat{\boldsymbol{\mu}}_{n'}, \widehat{\boldsymbol{\Sigma}}_{n'}))$ is computed by $\|\hat{\boldsymbol{\mu}}_n - \hat{\boldsymbol{\mu}}_{n'}\|_2^2 + \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}_n + \widehat{\boldsymbol{\Sigma}}_{n'} - 2(\widehat{\boldsymbol{\Sigma}}_n^{\frac{1}{2}}\widehat{\boldsymbol{\Sigma}}_{n'}\widehat{\boldsymbol{\Sigma}}_n^{\frac{1}{2}})^{\frac{1}{2}})$. Note that, we can compute the square root of the covariance matrix (i.e., $\widehat{\boldsymbol{\Sigma}}_n^{\frac{1}{2}}$) by Newton-Schulz algorithm (Muzellec and Cuturi 2018), making this Wasserstein distance differentiable.

[1] When $K > 2$, we can apply PCA to get the 2D ellipses. In particular, applying PCA to the matrix $\boldsymbol{U}$ constructed by all the $\hat{\boldsymbol{\mu}}$'s, we can obtain the positions of the 2D ellipses based on the coefficients of the top-2 principal components. For each ellipse, we apply PCA to the corresponding $\widehat{\boldsymbol{\Sigma}}$. The top-2 principal components and the corresponding coefficients determine its shape and size.

Regularized t-SNE Learning Framework

WatE learns the graph embedding model f_θ in a regularized t-SNE framework. This learning framework considers the following three objectives: i) preserve the original graphs' structural information in the latent space, ii) enhance the clustering structure of the embeddings, and iii) regularize the energy of the covariance matrices, respectively.

KL-divergence between the affinity matrices. Given the affinity matrix of the raw graphs and that of the graph embeddings, we learn our model by minimizing the KL-divergence between them, as the t-SNE method does:

$$\mathrm{KL}(\boldsymbol{P}(\mathcal{G})\|\boldsymbol{Q}(\theta)) = \langle \boldsymbol{P}, \log \boldsymbol{P}(\mathcal{G}) - \log \boldsymbol{Q}(\theta)\rangle, \tag{7}$$

where $\langle\cdot,\cdot\rangle$ represents the inner product of matrices. By minimizing the KL-divergence loss, we fit the model distribution to the data distribution, making the graph embeddings inherit the clustering structure of raw graphs.

Gromov-Wasserstein clustering loss. To further enhance the clustering structure of the graph embeddings, we introduce the Gromov-Wasserstein (GW) clustering loss (Chowdhury and Needham 2021; Gong, Nie, and Xu 2022) as a regularizer. Suppose that the graph embeddings belong to C clusters. The GW clustering loss of $\boldsymbol{Q}(\theta)$ corresponds to the following optimization problem:

$$\begin{aligned}
&\mathrm{L}_{\mathrm{gw}}\Big(\boldsymbol{Q}(\theta), \frac{1}{C}\boldsymbol{I}_C\Big) \\
&= \min_{\boldsymbol{T}\in\Pi(\frac{1}{N}\mathbf{1}_N, \frac{1}{C}\mathbf{1}_C)} \sum_{n,c,n',c'} \frac{\delta_{cc'}}{C} \log \frac{\delta_{cc'}}{C q_{nn'}(\theta)} t_{nc} t_{n'c'} \qquad (8)\\
&\Leftrightarrow \min_{\boldsymbol{T}\in\Pi(\frac{1}{N}\mathbf{1}_N, \frac{1}{C}\mathbf{1}_C)} -\mathrm{tr}(\boldsymbol{T}^T \log(\boldsymbol{Q}(\theta))\boldsymbol{T}),
\end{aligned}$$

where $\boldsymbol{I}_C$ is an identity matrix with size $C \times C$. $\delta_{cc'}$ is the Dirac function, which equals to 1 when $c = c'$ and 0 otherwise. We assume the empirical distribution of the graphs and that of the clusters are uniform. The matrix $\boldsymbol{T} = [t_{nc}] \in \Pi(\frac{1}{N}\mathbf{1}_N, \frac{1}{C}\mathbf{1}_C)$ is a joint distribution of the graphs and the clusters, whose element t_{nc} indicates the probability that the graph G_n belongs to the c-th cluster. As shown in (8), the GW clustering loss itself is an optimization problem, which minimizes the expectation of the KL-divergence $\mathrm{KL}(\frac{1}{C}\boldsymbol{I}_C\|\boldsymbol{Q}(\theta))$. Its formulation corresponds to the GW distance, in which the d_A is specified as the KL divergence. The end line of (8) shows that the GW clustering loss is equivalent to solving a generalized spectral clustering problem with a doubly-stochastic constraint (Chowdhury and Needham 2021; Gong, Nie, and Xu 2022). The GW loss in (8) can be solved via the conditional gradient algorithm in (Titouan et al. 2019).

Energy-based covariance regularizer. Finally, for all observed graphs, we regularize their covariance matrices by

$$\mathrm{R}(\theta) = \frac{1}{N}\sum\nolimits_{n=1}^{N} \Big\|\widehat{\boldsymbol{\Sigma}}_n(\theta) - \frac{1}{K}\boldsymbol{I}_K\Big\|_F^2, \tag{9}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of matrix. From the viewpoint of statistics, this regularizer i) makes the variance along each embedding dimension approach 1, and ii) penalizes the correlation across different embedding dimensions. In the following graph visualization experiments, we will show that this regularizer helps to improve the subjective visual effect of graph embeddings.

Taking the above three terms into account, we learn our graph embedding model by

$$\min_\theta \mathrm{KL}(\boldsymbol{P}\|\boldsymbol{Q}(\theta)) + \alpha \mathrm{L}_{\mathrm{gw}}\Big(\boldsymbol{Q}(\theta), \frac{1}{C}\boldsymbol{I}_C\Big) + \beta \mathrm{R}(\theta), \tag{10}$$

where the hyperparameters, α and β, control the significance of the GW loss and that of the covariance regularizer, respectively. Differing from the t-SNE in (1), which learns embeddings in a transductive way, our method learns an inductive embedding model f_θ. Therefore, when new graphs come, we can derive and visualize their embeddings directly based on the learned model. The algorithmic scheme is shown in Algorithm 1, and Newton-Schulz algorithm of $\widehat{\boldsymbol{\Sigma}}_n^{\frac{1}{2}}$ and conditional gradient algorithm of GW and FGW distance are presented at https://github.com/minjiecheng/WatE.

Algorithm 1: The Rt-SNE framework for WatE model

Require: Given a set of graphs $\mathcal{G} = \{G_n\}_{n=1}^N$
1: Compute the affinity matrix $\boldsymbol{P}(\mathcal{G})$ via (3) in advance.
2: **For** Epoch $= 1, 2, ...$
3: **For** A batch of graphs $\mathcal{G}_B \subset \mathcal{G}$
4: Compute Wasserstein distances, get $\boldsymbol{Q}_B(\theta)$ via (6).
5: Solve (8) and obtain $\boldsymbol{T}^*$.
6: Fix $\boldsymbol{T}^*$ as a constant and compute L_{gw} accordingly.
7: Compute the objective function in (10).
8: Update θ via Adam (Kingma and Ba 2015).
9: $\theta^* := \theta$.

Experiments

To demonstrate the effectiveness of WatE, we test it on various graph datasets and evaluate the graph visualization results on both subjective visual effects and objective clustering accuracy. We apply four TU graph datasets (Morris et al. 2020), including **IMDB-B** (social networks), **MUTAG**, **PTC-MR** (molecular datasets) and **AIDS** (biomedical graph datasets). The baselines consist of two OT-based graph factorization methods, i.e., **GWF** (Xu et al. 2023) and **GDL** (Vincent-Cuaz et al. 2021), and two GNN-based unsupervised embedding methods, i.e., **GraphCL** (You et al. 2020) and **ADGCL** (Suresh et al. 2021). For qualitative evaluation, we add five more kernel-based methods, i.e., **RWK** (Gärtner, Flach, and Wrobel 2003), **SPK** (Borgwardt and Kriegel 2005), **GK** (Shervashidze et al. 2009), **WLK** (Shervashidze et al. 2011), and **FGWK** (Titouan et al. 2019). Furthermore, we demonstrate the generalization power of our method and its robustness to hyperparameters.

Visual and Qualitative Comparisons

Information-enriched graph visualization. To validate the effectiveness of Information-enriched Graph Visualization, we selected the protein pre-training representation

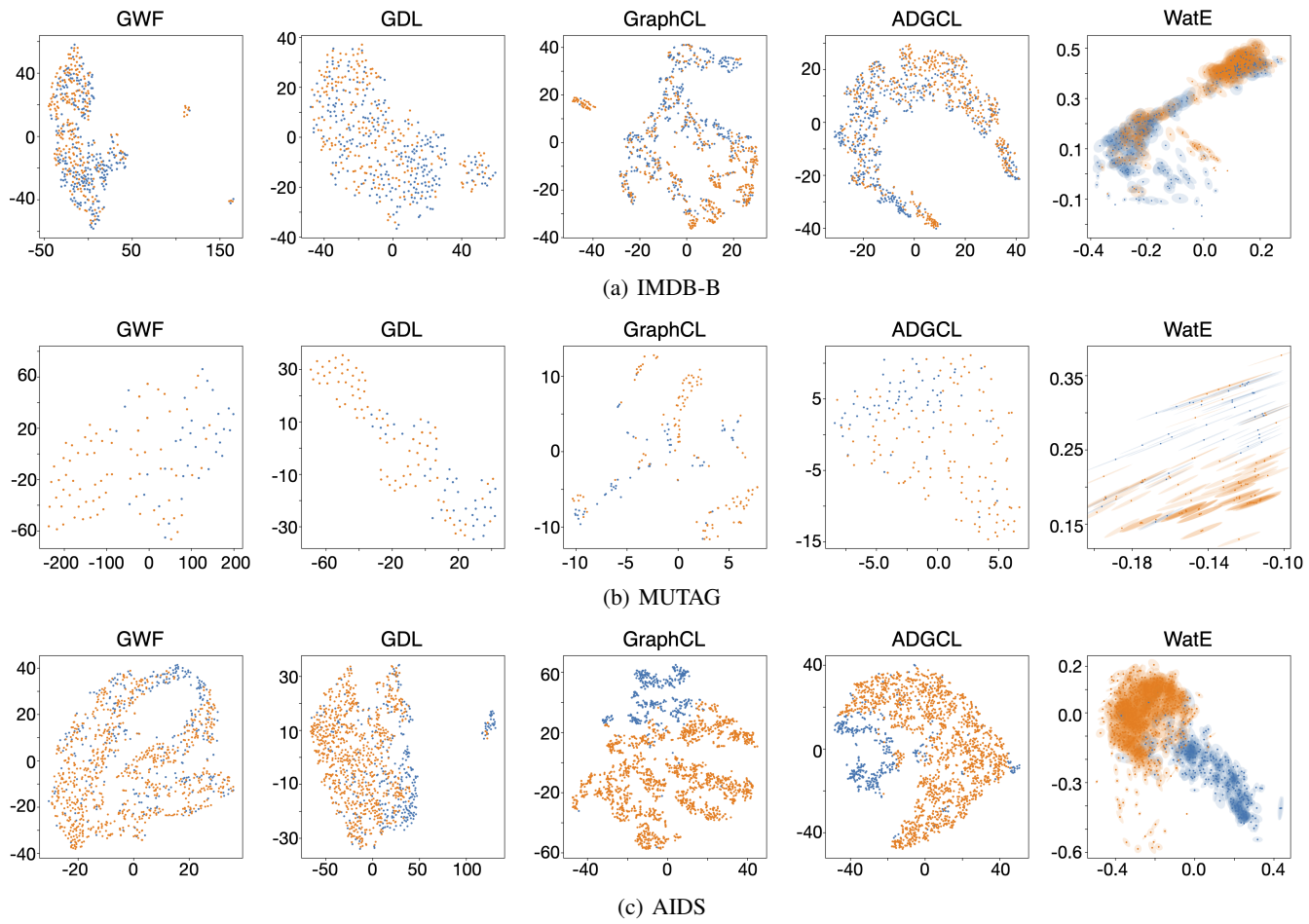


Figure 3: Comparing graph embedding methods on the IMDB-B, MUTAG, and AIDS datasets.

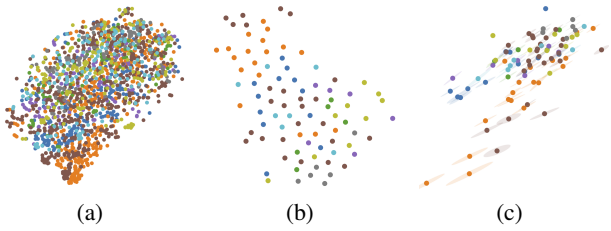


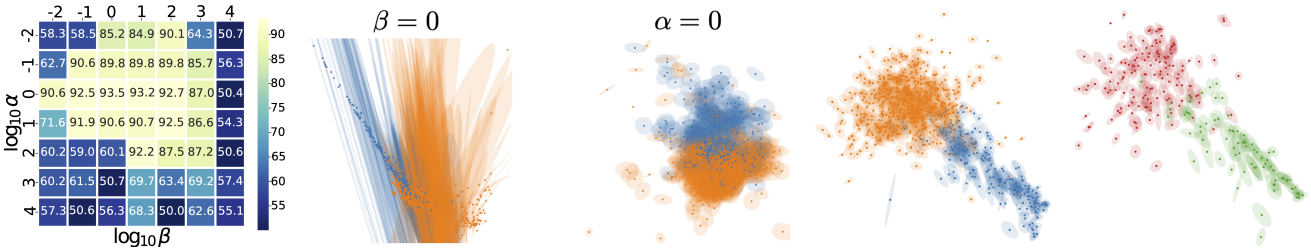
Figure 4: Illustrations of information-enriched graph visualization: (a) Protein residue embeddings via t-SNE, (b) Block embeddings via t-SNE, (c) Block embeddings via WatE. Colors indicate different proteins.

model GearNet (Zhang et al. 2022) for visualizing downstream proteins as a comparison. We plotted residue embeddings for 8 proteins in the Enzyme Commission (EC) number prediction task as node-level visualizations. Each protein was divided into blocks of 20 residues to create substructures. These block embeddings were visualized using t-SNE and WatE. Figure 4 shows the corresponding visualization results, including the visualization of residue embeddings and the t-SNE and WatE visualizations of block embeddings.

As shown in Figure 4, the node-level visualization appear chaotic, failing to provide useful information. The block-level visualization via t-SNE loses too much information, making it impossible to visualize the distribution of residues within each block. In addition to presenting the overall embedding between blocks, WatE also uses ellipses for visualization to display the distribution of residues within each block. This can reveal information contained within different blocks, such as similar substructures or functions. This approach may be valuable for protein screening and design.

Clustering graph visualization. We compare our WatE method with the baselines achieving explicit graph embeddings in clustering visualization tasks. The visualization results are shown in Figure 3. For our method, we set $K = 2$ and learn a graph embedding model accordingly. We apply the t-SNE method for the baselines to visualize their graph embeddings in the 2D plane. The proposed graph embeddings are superior to those of the baselines on visual effects.

Firstly, the visualization results obtained by our method have clear clustering structures consistently for IMDB-B,



(a) Impacts of α and β (b) Significance of regularizers (c) The generalization power of WatE on AIDS

Figure 5: (a, b) The ablation study on α and β. (c) Training on left, testing on right for each subfigure.

Method	K	IMDB-B	MUTAG	PTC-MR	AIDS
RWK	–	$49.95_{\pm 0.01}$	$65.84_{\pm 0.98}$	$50.53_{\pm 0.19}$	$55.90_{\pm 0.34}$
SPK	–	$50.41_{\pm 0.21}$	$66.07_{\pm 0.95}$	$50.52_{\pm 0.30}$	$67.54_{\pm 0.52}$
GK	–	$50.17_{\pm 0.12}$	$55.37_{\pm 1.70}$	$50.55_{\pm 0.32}$	$66.87_{\pm 0.45}$
WLK	–	$50.07_{\pm 0.07}$	$58.14_{\pm 1.25}$	$50.36_{\pm 0.26}$	$62.59_{\pm 6.10}$
FGWK	–	$49.98_{\pm 0.20}$	$51.05_{\pm 0.79}$	$50.33_{\pm 0.43}$	$74.20_{\pm 1.54}$
GWF	16	$51.18_{\pm 0.06}$	$67.92_{\pm 1.21}$	$51.48_{\pm 0.60}$	$85.39_{\pm 2.82}$
GDL	16	$51.40_{\pm 0.48}$	$70.18_{\pm 0.23}$	$\mathbf{51.64_{\pm 0.60}}$	$86.80_{\pm 1.75}$
GraphCL	96	$52.33_{\pm 0.36}$	$65.62_{\pm 2.42}$	$50.24_{\pm 0.14}$	$88.16_{\pm 6.22}$
ADGCL	160	$52.45_{\pm 0.30}$	$69.74_{\pm 0.04}$	$49.87_{\pm 0.09}$	$92.55_{\pm 0.85}$
WatE	2	$52.42_{\pm 1.71}$	$69.49_{\pm 0.96}$	$50.19_{\pm 0.18}$	$91.14_{\pm 0.48}$
	8	$\mathbf{52.47_{\pm 0.64}}$	$\mathbf{71.91_{\pm 1.98}}$	$50.55_{\pm 0.24}$	$\mathbf{93.21_{\pm 2.47}}$

Table 1: Comparisons for various embedding methods on their clustering performance (Rand Index $\pm$ std. (%)). K equals "–" means not deriving graph embeddings explicitly.

MUTAG, and AIDS, which are visually better than the baselines. Secondly, while the baselines represent each graph as a single point, our WatE method visualizes it as an ellipse based on the mean and covariance of the node embeddings, which preserves more node-level information in the visualization results than the baselines. For example, besides showing the clustering structure of the MUTAG graphs, our method further indicates that i) each graph's node embeddings are scattered in an anisotropic way, and ii) the distributions of different graphs' node embeddings are similar (i.e., the top-1 eigenvectors of the covariance matrices are similar). These experimental results demonstrate that our WatE method provides two perspectives (i.e., the mean and covariance of node embeddings) for graph analysis, achieving semantically-meaningful graph visualization.

Quantitative Comparisons

For each method, we test it in 10 trials and record the mean and the standard deviation of its clustering results on each graph dataset. The quantitative clustering results of different methods are shown in Table 1. For our WatE method, when setting $K = 2$, we can derive the mean of each graph's node embeddings has two dimension and the degree of freedom of covariance matrix is six. Therefore, we can cluster observed graphs i) merely based on their 2D mean vectors and ii) by eight-dimensional latent vectors consisting of mean and covariance information, respectively. Compared to the baselines, especially the GNN-based methods (GraphCL and ADGCL), WatE embeds graphs with much lower dimensions and obtains competitive clustering results. When considering the mean and covariance information jointly (i.e., $K = 8$), WatE achieves the best clustering results on three of four datasets. For PTC-MR, our method's performance is limited because some graphs have different labels but similar structures. Consequently, the FGW distance between graphs does not always reflect label differences, resulting in a noisy affinity matrix $\boldsymbol{P}$ and learned $\boldsymbol{Q}$. Improving robustness to data noise will be a key focus of our future research.

Analytic experiments

Impacts of α and β. As shown in (10), α and β control the significance of the GW loss and that of the covariance regularizer, respectively. We set $\alpha = 1$ and $\beta = 100$ by default. In Figure 5(a), when $0.1 \leq \alpha \leq 10$ and $1 \leq \beta \leq 100$, the performance of WatE is stable, demonstrating its robustness to hyperparameters. In Figure 5(b), when $\beta = 0$, we learn the model without the covariance regularizer. In such a situation, the visualization result is unsatisfactory — the energy of covariance matrices is unconstrained. Thus we cannot preserve the concentricity of the node embeddings within each graph during training. On the contrary, when $\alpha = 0$, we learn the model without the GW loss, and the clustering structure of the learned embeddings is not so significant as that under the proposed setting.

Generalization power. As aforementioned, our WatE is an inductive graph visualization method. After learning the graph embedding model, we gain the capability to derive and visualize the embeddings of new graphs directly. Figure 5(c) shows the generalization power of our method. For each dataset, we train our model on 80% of the graphs and then visualize the entire dataset. We find that the distribution of the remaining 20% of graphs is similar to the training graphs, even though they were not seen during training.

Conclusion and Future Work

We propose an information-enriched graph visualization method with the help of optimal transport techniques, learning an inductive graph embedding model within a regularized t-SNE framework. This approach is suitable for graph visualization and clustering tasks. Utilizing the information from ellipses to discover protein structures and functions will be a focus of our future research.

Acknowledgments

This work was supported by National Natural Science Foundation (62106271, 92270110), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China. We also acknowledge the support provided by the fund for building world-class universities (disciplines) of Renmin University of China and by the funds from Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, and from Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4): 433–459.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Bachmann, F.; Hennig, P.; and Kobak, D. 2022. Wasserstein t-SNE. *arXiv preprint arXiv:2205.07531*.

Borgwardt, K. M.; and Kriegel, H.-P. 2005. Shortest-path kernels on graphs. In *IEEE International Conference on Data Mining*, 8–pp.

Chowdhury, S.; and Mémoli, F. 2019. The gromov–wasserstein distance between networks and stable network invariants. *Information and Inference: A Journal of the IMA*, 8(4): 757–787.

Chowdhury, S.; and Needham, T. 2021. Generalized spectral clustering via Gromov-Wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*, 712–720. PMLR.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Gärtner, T.; Flach, P.; and Wrobel, S. 2003. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, 129–143. Springer.

Gong, F.; Nie, Y.; and Xu, H. 2022. Gromov-Wasserstein multi-modal alignment and clustering. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 603–613.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kondor, R.; and Pan, H. 2016. The multiscale laplacian graph kernel. In *Advances in Neural Information Processing Systems*, 2990–2998.

Mémoli, F. 2011. Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4): 417–487.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*.

Muzellec, B.; and Cuturi, M. 2018. Generalizing point embeddings using the wasserstein space of elliptical distributions. *Advances in Neural Information Processing Systems*, 31.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

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.; and Solomon, J. 2016. Gromov-Wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, 2664–2672. PMLR.

Rippl, T.; Munk, A.; and Sturm, A. 2016. Limit laws of the empirical Wasserstein distance: Gaussian distributions. *Journal of Multivariate Analysis*, 151: 90–109.

Shervashidze, N.; Schweitzer, P.; Leeuwen, E. J. v.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep): 2539–2561.

Shervashidze, N.; Vishwanathan, S.; Petri, T.; Mehlhorn, K.; and Borgwardt, K. 2009. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, 488–495.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2019. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500): 2319–2323.

Titouan, V.; Courty, N.; Tavenard, R.; and Flamary, R. 2019. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, 6275–6284. PMLR.

Tolstikhin, I.; Bousquet, O.; Gelly, S.; and Schoelkopf, B. 2018. Wasserstein Auto-Encoders. In *International Conference on Learning Representations*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Villani, C. 2009. *Optimal transport: old and new*, volume 338. Springer.

Vincent-Cuaz, C.; Vayer, T.; Flamary, R.; Corneli, M.; and Courty, N. 2021. Online graph dictionary learning. In *International Conference on Machine Learning*, 10564–10574. PMLR.

Wang, H.; and Banerjee, A. 2014. Bregman alternating direction method of multipliers. *Advances in Neural Information Processing Systems*, 27.

Xie, Y.; Wang, X.; Wang, R.; and Zha, H. 2020. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, 433–453. PMLR.

Xu, H.; Liu, J.; Luo, D.; and Carin, L. 2023. Representing Graphs via Gromov-Wasserstein Factorization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(01): 999–1016.

Xu, H.; Luo, D.; and Carin, L. 2019. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823.

Zhang, Z.; Xu, M.; Jamasb, A.; Chenthamarakshan, V.; Lozano, A.; Das, P.; and Tang, J. 2022. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*.