Generative Graph Dictionary Learning

Zhichen Zeng [1] **Ruike Zhu** [1] **Yinglong Xia** [2] **Hanqing Zeng** [2] **Hanghang Tong** [1]

Abstract

Dictionary learning, which approximates data samples by a set of shared atoms, is a fundamental task in representation learning. However, dictionary learning over graphs, namely graph dictionary learning (GDL), is much more challenging than vectorial data as graphs lie in disparate metric spaces. The sparse literature on GDL formulates the problem from the reconstructive view and often learns linear graph embeddings with a high computational cost. In this paper, we propose a Fused Gromov-Wasserstein (FGW) Mixture Model named FRAME to address the GDL problem from the generative view. Equipped with the graph generation function based on the radial basis function kernel and FGW distance, FRAME generates nonlinear embedding spaces, which, as we theoretically proved, provide a good approximation of the original graph spaces. A fast solution is further proposed on top of the expectation-maximization algorithm with guaranteed convergence. Extensive experiments demonstrate the effectiveness of the obtained node and graph embeddings, and our algorithm achieves significant improvements over the state-of-the-art methods.

1. Introduction

In the era of big data and AI, graphs originate from various domains carrying rich information. Finding low-dimensional representations encoding graph structural information, i.e., graph representation learning, is the key stepping stone behind various graph-based applications in bioinformatics (Ktena et al., 2017), chemistry (Jin et al., 2017), social networks (Yanardag & Vishwanathan, 2015), and many more.

Dictionary learning, which seeks for low-dimensional representations for data samples based on a set of shared patterns, namely atoms, has achieved great success in vectorial data. However, graph dictionary learning (GDL) is much more challenging as graphs lie in disparate spaces (Peyré et al., 2016; Xu, 2020). Thanks to the recent advancement of optimal transport (OT), together with powerful graph distance measures based on the Wasserstein-like distances (Mémoli, 2011; Sturm, 2012; Titouan et al., 2019), a few GDL methods (Xu, 2020; Vincent-Cuaz et al., 2022; 2021; Liu et al., 2023) have been recently proposed. Most of the existing GDL methods follow a reconstructive formulation by minimizing the Wasserstein-like distances between the original and reconstructed graphs, but may suffer from several fundamental limitations. First (linear embedding), graphs are approximated by the linear combination of atoms, which in turn leads to linear embeddings with limited representation power. Second (single-level embedding), the rich information in the OT coupling is rarely exploited by existing GDL methods to generate embeddings at other levels (e.g., node or subgraph level) with quantitatively identified cross-level relationships. Third (computation), although stochastic gradient descent (SGD) provides a scalable approach for large graphs, other accompanied optimization steps along SGD still bear intensive computation.

Contributions. In this paper, we propose a novel GDL method named FRAME to address the aforementioned limitations. Using the graph generation function based on the Fused Gromov-Wasserstein (FGW) distance and the radius basis function (RBF) kernel, the GDL problem is formulated from the generative perspective by maximizing the likelihood of generating graphs from atoms, through which the nonlinear graph-atom relationship can be captured. Besides, by utilizing the accompanied node correspondence information, the proposed method can jointly generate graph, subgraph and node embeddings. A fast solution based on the expectation-maximization (EM) algorithm is further proposed achieving quadratic time complexity with guaranteed convergence. Theoretical analysis shows that FRAME generates a low-dimensional embedding space that well-approximates the original graph space with little information loss. Extensive experiments show that FRAME achieves significant improvement on graph-level and node-level tasks, outperforming the state-of-the-art by 8.0% on graph classification, 0.5% on graph clustering, and 2.5% on

[1] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, USA [2] Meta, CA, USA. Correspondence to: Hanghang Tong <htong@illinois.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

node clustering, respectively.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries and formulates the GDL problem. Section 3 presents the proposed FRAME and provides relevant theoretical analyses. Extensive experiments are carried out in Section 4. Related works and conclusions are given in Sections 5 and 6 respectively.

2. Problem Formulation

2.1. Notations

We use bold uppercase letters for matrices (e.g., $\mathbf{A}$), bold lowercase letters for vectors (e.g., $\mathbf{s}$), calligraphic letters for sets (e.g., $\mathcal{G}$), and lowercase letters for scalars (e.g., α). The element (i, j) of a matrix $\mathbf{A}$ is denoted as $\mathbf{A}(i, j)$. The transpose of $\mathbf{A}$ is denoted by the superscript $\intercal$ (e.g., $\mathbf{A}^\intercal$). An attributed graph is represented as $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ where $\mathbf{A}$ and $\mathbf{X}$ denote the adjacency matrix and the node attribute matrix respectively. The simplex histogram with n bins is denoted as $\Delta_n = \{\boldsymbol{\mu} \in \mathbb{R}_n^+ | \sum_{i=1}^n \boldsymbol{\mu}(i) = 1\}$. The probabilistic coupling is denoted as $\Pi(\cdot, \cdot)$, and the inner product is denoted as $\langle \cdot, \cdot \rangle$. For simplicity, we denote the set of positive integers no greater than n as $\mathbb{N}_{\leq n}^+$.

2.2. Optimal Transport on Graphs

The OT theory seeks for the optimal coupling between distributions and has achieved great success in various graph-related tasks including graph comparison (Maretic et al., 2019; Titouan et al., 2019), graph matching (Xu et al., 2019b; Zeng et al., 2023), and graph representation learning (Kolouri et al., 2021). To adopt the OT theory for graphs, a graph is represented as a probability measure on the product space of graph structure and node attributes, where elements indicate the importance of nodes. Without any prior knowledge on nodes, uniform node importance is often the default choice and a graph $\mathcal{G}$ with n nodes is represented as a uniform histogram $\boldsymbol{\mu} = \frac{\mathbf{1}_n}{n}$ (Vincent-Cuaz et al., 2021).

Definition 2.1. Fused Gromov-Wasserstein distance (Peyré et al., 2016; 2019; Titouan et al., 2019).
Given two graphs $\mathcal{G}_1, \mathcal{G}_2$ represented by probability measures $\boldsymbol{\mu}_1 = \sum_{i=1}^{n_1} h_i \delta_{x_i, \mathbf{X}_1(x_i)}, \boldsymbol{\mu}_2 = \sum_{j=1}^{n_2} g_j \delta_{y_j, \mathbf{X}_2(y_j)}$, where $h \in \Delta_n, g \in \Delta_m$ are histograms, a cross-graph matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ measuring cross-graph node distances based on attributes, and two intra-graph matrices $\mathbf{C}_1 \in \mathbb{R}^{n_1 \times n_1}, \mathbf{C}_2 \in \mathbb{R}^{n_2 \times n_2}$ measuring intra-graph node similarity based on graph structure, the q-FGW distance $\text{FGW}_{q,\alpha}(\mathcal{G}_1, \mathcal{G}_2)$ is defined as:

$$\min_{\mathbf{S} \in \Pi(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)} \sum_{\substack{x_1 \in \mathcal{G}_1 \\ y_1 \in \mathcal{G}_2}} (1-\alpha) \mathbf{M}(x_1, y_1)^q \mathbf{S}(x_1, y_1) + \sum_{\substack{x_1, x_2 \in \mathcal{G}_1 \\ y_1, y_2 \in \mathcal{G}_2}} \alpha |\mathbf{C}_1(x_1, x_2) - \mathbf{C}_2(y_1, y_2)|^q \mathbf{S}(x_1, y_1) \mathbf{S}(x_2, y_2) \tag{1}$$

where q and α are the order of FGW distance and weight hyperparameters respectively. Intuitively, Eq. (1) measures the minimum effort of transporting $\mathcal{G}_1$ to $\mathcal{G}_2$ based on the node attribute dissimilarities $\mathbf{M}(x_1, y_1)$ and structure dissimilarities $\mathbf{C}_1(x_1, x_2) - \mathbf{C}_2(y_1, y_2)$. For simplicity, we omit the subscript and use $\text{FGW}(\mathcal{G}_1, \mathcal{G}_2)$ throughout the paper.

The FGW distance serves as a powerful graph distance measure as both node attributes and graph structure are exploited (Titouan et al., 2019). Based on Definition 2.1, the FGW barycenter problem is defined as follows:

Definition 2.2. Fused Gromov-Wasserstein barycenter (Titouan et al., 2019).
Given N graphs $\mathcal{G}_i$ with intra-graph matrices $\mathbf{C}_{\mathcal{G}_i}$ and weights λ_i for $i \in \mathbb{N}_{\leq N}^+$ satisfying $\sum_{i=1}^N \lambda_i = 1$, the FGW barycenter problem aims to find the barycenter graph $\mathcal{B}$ with the intra-graph matrix $\mathbf{C}_\mathcal{B}$ and node attribute matrix $\mathbf{X}_\mathcal{B}$ such that the weighted sum of FGW distances between $\mathcal{G}_i$ and $\mathcal{B}$ is minimized:

$$\mathbf{C}_\mathcal{B}, \mathbf{X}_\mathcal{B} = \underset{\mathbf{C}_\mathcal{B}, \mathbf{X}_\mathcal{B}}{\arg\min} \sum_{i=1}^N \lambda_i \text{FGW}(\mathcal{G}_i, \mathcal{B}). \tag{2}$$

In the GDL problem, we are interested in finding multiple barycenters $\mathcal{B}_1, \ldots, \mathcal{B}_K$, namely atoms, from the input graphs $\mathcal{G}_1, \ldots, \mathcal{G}_N$. For clarity, we slightly abuse the subscript i to denote graph $\mathcal{G}_i$ and k to denote atom $\mathcal{B}_k$ exclusively (e.g., $\mathbf{X}_i = \mathbf{X}_{\mathcal{G}_i}, \mathbf{X}_k = \mathbf{X}_{\mathcal{B}_k}$). Following a similar approach as (Titouan et al., 2019), the cross-graph and intra-graph matrices are defined as follows. We use the L_2 norm between the attributes of node $x \in \mathcal{G}_i$ and $y \in \mathcal{B}_k$ as cross-graph matrices $\mathbf{M}_{i,k}(x, y)$ and the adjacency matrices $\mathbf{A}_i$ as the intra-graph matrices $\mathbf{C}_i$.

2.3. Generative Graph Dictionary Learning

Unlike the existing methods formulating the problem from the reconstructive perspective, we formulate the GDL problem from the generative perspective as follows:

Definition 2.3. Generative graph dictionary learning.
Given N graphs $\mathcal{G}_i$ for $i \in \mathbb{N}_{\leq N}^+$ and a graph generation function $p(\mathcal{G}_i | \mathcal{B}_k)$ indicating the probability of generating graph $\mathcal{G}_i$ from atom $\mathcal{B}_k$. The generative graph dictionary learning problem aims to find K attributed atoms $\mathcal{B}_k$ and corresponding prior probability $\pi_k = p(\mathcal{B}_k)$, such that the log likelihood of generating $\mathcal{G}_i$ from $\mathcal{B}_k$ for $i \in \mathbb{N}_{\leq N}^+, k \in \mathbb{N}_{\leq K}^+$ is maximized. Denoting model parameters as $\Theta = \{\mathcal{B}_1, \ldots, \mathcal{B}_K, \pi_1, \ldots, \pi_K\}$, the objective function of generative graph dictionary learning is formulated as:

$$\Theta = \arg\max_\Theta \log p(\mathcal{G}_1, \ldots, \mathcal{G}_N | \Theta), \quad \text{s.t.} \sum_{k=1}^K \pi_k = 1. \tag{3}$$

Table 1. Comparison with existing GDL methods.

Method	Desired Property			
	nonlinear	multi-level	size-free	supervision
GWF (Xu, 2020)	✓	✗	✓	✗
srGW (Vincent-Cuaz et al., 2022)	✗	✗	✓	✗
GDL (Vincent-Cuaz et al., 2021)	✗	✗	✗	✗
RGWD (Liu et al., 2023)	✗	✗	✗	✗
FraMe (proposed)	✓	✓	✓	✓

In this paper, we assume graphs are i.i.d. data samples generated by the mixture of atoms (i.e., $p(\mathcal{G}_1, \dots, \mathcal{G}_N|\Theta) = \prod_{i=1}^{N} p(\mathcal{G}_i|\Theta)$), and a graph $\mathcal{G}_i$ is more likely to be generated by an atom $\mathcal{B}_k$ with a smaller $\text{FGW}(\mathcal{G}_i, \mathcal{B}_k)$ following the RBF kernel-based graph generation function as follows:

$$p(\mathcal{G}_i|\mathcal{B}_k) = \exp(-\sigma \text{FGW}(\mathcal{G}_i, \mathcal{B}_k)), \tag{4}$$

where $\sigma > 0$ is the length-scale parameter of the RBF kernel.

Before presenting our solution to the generative GDL problem in the next section, let us first summarize a few key desired properties of GDL. First (*P1. multi-level embedding*), the existing GDL methods almost exclusively focus on producing graph level embedding, while the rich node-level information in the accompanied OT coupling is rarely exploited. Ideally, the learned atoms should be able to generate multi-level embedding with quantitatively identified relationship between different levels (Du & Tong, 2019). This will not only help discover cross-level correlations, but also broaden the applicability of GDL to support node level or subgraph level learning tasks (e.g., node classification, link prediction, etc.). Second (*P2. nonlinear embedding*), most of the existing GDL methods generate linear embedding, which limits its representation power. It is desirable to generate nonlinear embedding without increasing its computational complexity. Third (*P3. size-free*), most of the reconstructive formulation (Vincent-Cuaz et al., 2021; Liu et al., 2023) learns atoms with the same size, but it is preferred to generate atoms with different sizes to capture features at multiple scales. Forth (*P4. incorporate supervision*), existing GDL methods only focus on the unsupervised setting, while utilizing supervision from labelled data may significantly enhance the GDL performance. Table 1 summarizes and compares the existing GDL methods. As we can see, the proposed FraMe is the only method that enjoys all these four desired properties.

3. Algorithm and Analysis

In this section, we introduce and analyze the proposed FraMe. The algorithm is first introduced in Section 3.1. Relevant analysis on embedding quality, convergence, and complexity are presented in Section 3.2. Further discussion and variants of FraMe are carried out in Appendix B.

3.1. FraMe Algorithm

In this subsection, we present our proposed algorithm FraMe. The overall optimization framework to maximize Eq. (3) can be divided into the expectation and maximization steps. By leveraging the posterior probability and optimal coupling, we can generate embeddings at multiple levels and reconstruct the origin graphs from the embedding space.

Expectation. We first introduce a set of latent variables $z_{i,k}$ indicating whether the graph $\mathcal{G}_i$ is generated from the atom $\mathcal{B}_k$. In the expectation step, we focus on computing the posterior probability $\gamma_{i,k} = p(z_{i,k} = 1|\mathcal{G}_i, \Theta)$, i.e., the probability of $\mathcal{G}_i$ being generated by $\mathcal{B}_k$ from the mixture model. For the t-th iteration, fixing atoms $\mathcal{B}_k^{(t-1)}$ and corresponding prior probability $\pi_k^{(t-1)}$, the posterior probability $\gamma_{i,k}^{(t)}$ can be computed based on the Bayes rule as follows:

$$\gamma_{i,k}^{(t)} = \frac{\pi_k^{(t-1)} p\left(\mathcal{G}_i|\mathcal{B}_k^{(t-1)}\right)}{\sum_{j=1}^{K} \pi_j^{(t-1)} p\left(\mathcal{G}_i|\mathcal{B}_j^{(t-1)}\right)}. \tag{5}$$

Maximization. Equipped with posterior probabilities $\gamma_{i,k}^{(t)}$ from the expectation step, the objective function in Eq. (3) is lower bounded by the following complete log-likelihood (Bishop & Nasrabadi, 2006):

$$\begin{aligned} \mathcal{Q}^{(t)}(\Theta) &= \sum_{i=1}^{N}\sum_{k=1}^{K} p\left(z_{i,k}|\mathcal{G}_i,\Theta^{(t-1)}\right) \log p(\mathcal{G}_i, z_{i,k}|\Theta) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{i,k}^{(t)} \left[\log \pi_k - \sigma \text{FGW}(\mathcal{G}_i, \mathcal{B}_k)\right]. \end{aligned} \tag{6}$$

In the maximization step, we focus on re-estimating model parameters to maximize the above complete log-likelihood.

First (maximization w.r.t. π_k), fixing the posterior probability $\gamma_{i,k}^{(t)}$ and the atom $\mathcal{B}_k^{(t-1)}$, the prior probability $\pi_k^{(t)}$ can be calculated by

$$\pi_k^{(t)} = \frac{\sum_{i=1}^{N} p\left(z_{i,k}=1|\mathcal{G}_i,\Theta^{(t-1)}\right)}{N} = \frac{\sum_{i=1}^{N} \gamma_{i,k}^{(t)}}{N}. \tag{7}$$

Second (maximization w.r.t. $\mathcal{B}_k = \{\mathbf{A}_k, \mathbf{X}_k\}$), fixing the posterior probability $\gamma_{i,k}^{(t)}$ and the prior probability $\pi_k^{(t)}$, the maximization w.r.t. $\mathcal{B}_k$ in Eq. (6) can be formulated as

$$\min_{\mathcal{B}_k} \sum_{i=1}^{N} \gamma_{i,k}^{(t)} \mathrm{FGW}(\mathcal{G}_i, \mathcal{B}_k). \tag{8}$$

Note that the above optimization problem corresponds to the FGW barycenter problem in Eq. (2) and can be efficiently solved by the block coordinate descent (BCD) algorithm (Ferradans et al., 2014; Titouan et al., 2019). Specifically, the objective function in Eq. (8) is minimized w.r.t. the optimal coupling $\mathbf{S}_{i,k}$ between $\mathcal{G}_i$ and $\mathcal{B}_k$, atom adjacency matrix $\mathbf{A}_k$ and atom attribute matrix $\mathbf{X}_k$ iteratively. For the l-th iteration, the minimization w.r.t. three variables are computed as follows.

(1) Fixing $\mathbf{A}_k^{(l-1)}$ and $\mathbf{X}_k^{(l-1)}$, the objective function in Eq. (8) is equivalent to calculating (NK) FGW distances independently. For each FGW distance, a fast solution based on conditional gradient (CG) (Jaggi, 2013) is proposed by (Titouan et al., 2019) with convergence to a stationary point of the non-convex problem (Lacoste-Julien, 2016). Each CG subproblem is formulated as:

$$\begin{aligned} &\min_{\mathbf{S}\in\Pi(\boldsymbol{\mu}_i,\boldsymbol{\mu}_k)} \langle \mathbf{G}_{i,k}^{(l)}, \mathbf{S} \rangle, \\ &\mathbf{G}_{i,k}^{(l)} = \nabla_{\mathbf{S}} \mathrm{FGW} = (1-\alpha)\mathbf{M}_{i,k}^{(l-1)} + 2\alpha \mathbf{L}_{i,k}^{(l-1)} \otimes \mathbf{S}, \end{aligned} \tag{9}$$

where $\otimes$ is the Kronecker product, and $\mathbf{L}_{i,k}^{(l-1)}$ is a 4-dimensional tensor with $\mathbf{L}_{i,k}^{(l-1)}(x_1, x_2, y_1, y_2) = |\mathbf{A}_i(x_1, x_2) - \mathbf{A}_k^{(l-1)}(y_1, y_2)|^q$ for $x_1, x_2 \in \mathcal{G}_i$ and $y_1, y_2 \in \mathcal{B}_k$. Note that each subproblem in Eq. (9) corresponds to an OT problem, which can be efficiently solved by the inexact proximal point method with guaranteed convergence to the global optimum (Xie et al., 2020; Xu et al., 2019b).

(2) Fixing $\mathbf{S}_{i,k}^{(l)}$ and $\mathbf{X}_k^{(l-1)}$, the optimal intra-graph matrix $\mathbf{A}_k$ can be computed based on the first-order optimality condition as follows (Peyré et al., 2016):

$$\mathbf{A}_k^{(l)} = \frac{\sum_{i=1}^{N} \gamma_{i,k} \mathbf{S}_{i,k}^{(l)\mathsf{T}} \mathbf{A}_i \mathbf{S}_{i,k}^{(l)}}{\boldsymbol{\mu}_k \boldsymbol{\mu}_k^T}. \tag{10}$$

(3) Fixing $\mathbf{S}_{i,k}^{(l)}$ and $\mathbf{A}_k^{(l)}$, the objective function in Eq. (8) is quadratic w.r.t. $\mathbf{X}_k^{(l)}$, whose optimal solution can be computed as follows (Cuturi & Doucet, 2014):

$$\mathbf{X}_k^{(l)} = \mathrm{diag}\left(\frac{1}{\boldsymbol{\mu}_k}\right) \sum_{i=1}^{N} \gamma_{i,k} \mathbf{S}_{i,k}^{(l)} \mathbf{X}_i. \tag{11}$$

Combining Eqs. (5)-(11), the overall algorithm of FRAME is given in Algorithm 1.

Algorithm 1 FRAME

```
Input: $N$ graphs $\mathcal{G}_i = \{\mathbf{A}_i, \mathbf{X}_i\}$, number of nodes in $K$ atoms $n_k$, hyperparameters $\alpha, q, T, L$.
Randomly initialize atoms $\pi_k^{(0)}, \mathbf{A}_k^{(0)}, \mathbf{X}_k^{(0)}$;
Initialize marginal dist. $\boldsymbol{\mu}_i = \frac{\mathbf{1}_{n_i}}{n_i}, \boldsymbol{\mu}_k = \frac{\mathbf{1}_{n_k}}{n_k}$;
for $t \in \mathbb{N}^+_{\leq T}$ do
    for $i \in \mathbb{N}^+_{\leq N}$ and $k \in \mathbb{N}^+_{\leq K}$ do
        Update cross and intra-graph matrices $\mathbf{M}_{i,k}^{(t)}, \mathbf{L}_{i,k}^{(t)}$;
        Update posterior probability $\gamma_{i,k}^{(t)}$ by Eq. (5);
    end for
    for $k \in \mathbb{N}^+_{\leq K}$ do
        Update prior probability $\pi_k^{(t)}$ by Eq. (7);
        Update $\mathbf{S}_{i,k}^{(t)}, \mathbf{A}_k^{(t)}, \mathbf{X}_k^{(t)}$ by running $L$ conditional gradient iterations in Eqs. (9)-(11);
    end for
end for
return posterior probability $\gamma_{i,k}^{(T)}$, optimal coupling $\mathbf{S}_{i,k}^{(T)}$, learned atoms $\mathcal{B}_k^{(T)} = \{\mathbf{A}_k^{(T)}, \mathbf{X}_k^{(T)}\}$.
```

Semi-supervised FRAME. Existing GDL methods (Xu, 2020; Vincent-Cuaz et al., 2021; 2022; Liu et al., 2023) solely focus on the unsupervised setting, while it would be beneficial to incorporate external supervision when a small portion of graph labels are available. FRAME can naturally incorporate such supervision based on the semi-supervised EM algorithm (Nigam et al., 2000) to derive the semi-supervised variant named ss-FRAME. Specifically, in the semi-supervised setting, we are given N_u unlabelled graphs $\mathcal{G}_1, \cdots, \mathcal{G}_{N_u}$ and N_l labelled graphs $\mathcal{G}_{N_u+1}, \cdots, \mathcal{G}_{N_u+N_l}$ with labels $y_{N_u+1}, \cdots, y_{N_u+N_l}$. Following a similar derivation in the unsupervised setting, we can still adopt the maximization step in Eqs. (7)-(11) but with a modified expectation step as

$$\gamma_{i,k}^{(t)} = \begin{cases} \dfrac{\pi_k^{(t-1)} p\left(\mathcal{G}_i | \mathcal{B}_k^{(t-1)}\right)}{\sum_{j=1}^{K} \pi_j^{(t-1)} p\left(\mathcal{G}_i | \mathcal{B}_j^{(t-1)}\right)}, \text{ if } \mathcal{G}_i \text{ is unlabelled} \\ 1, \text{if } \mathcal{G}_i \text{ is labelled and } y_i = k \\ 0, \text{if } \mathcal{G}_i \text{ is labelled and } y_i \neq k \end{cases}$$

Towards Nonlinear Embeddings. Different from existing GDL methods that generate the linear representation of graphs based on the atoms (Vincent-Cuaz et al., 2022; 2021; Liu et al., 2023), the proposed FRAME generates nonlinear embeddings at node, subgraph and graph levels.

For node embeddings, by regarding nodes in the atoms as the bases of the node embedding space and using the optimal coupling $\mathbf{S}_{i,k}(v_p, v_q)$ as the coordinate of node $v_p \in \mathcal{G}_i$ w.r.t. the basis $v_q \in \mathcal{B}_k$, the node embedding $\mathbf{z}_{v_p}$ of $v_p \in \mathcal{G}_i$ is the concatenation of the optimal couplings as follows:

$$\mathbf{z}_{v_p} = [\gamma_{i,1} \mathbf{S}_{i,1}(v_p, :) \| \cdots \| \gamma_{i,K} \mathbf{S}_{i,K}(v_p, :)], \tag{12}$$

Intuitively, $\mathbf{S}_{i,k}(v_p, v_q)$ indicates the conditional probability of nodes $v_p \in \mathcal{G}_i$ and $v_q \in \mathcal{B}_k$ on that $\mathcal{G}_i$ is generated from $\mathcal{B}_k$, i.e., $p(v_p, v_q|z_{i,k} = 1)$. Therefore, each element in node embedding $\mathbf{z}_{v_p}$ corresponds to the marginal probability of v_p, v_q, that is

$$\begin{aligned} p(v_p, v_q) =& \sum_{z_{i,k}} p(v_p, v_q|z_{i,k})p(z_{i,k}) \\ =& \gamma_{i,k}\mathbf{S}_{i,k}(v_p, v_q) \end{aligned} , \tag{13}$$

Similarly, we can generate graph embeddings by measuring the joint probability of graph $\mathcal{G}_i$ and atom $\mathcal{B}_k$, i.e., $p(\mathcal{G}_i, \mathcal{B}_k)$. To be specific, we perform a linear transformation on node embeddings in Eq. (13) as follows:

$$\mathbf{w}_i = \sum_{v_p \in \mathcal{G}_i} \mathbf{z}_{v_p}\mathbf{W},$$

where $\mathbf{W} \in \mathbb{R}^{\sum_{i=1}^{K} n_{\mathcal{B}_k} \times K}$ is the indicator matrix such that each element $\mathbf{W}(v_q, k)$ indicates whether the atom node v_q belongs to the atom $\mathcal{B}_k$ as follows:

$$\mathbf{W} = \begin{bmatrix} \mathbf{1}_{n_{\mathcal{B}_1}} & \mathbf{0}_{n_{\mathcal{B}_1}} & \cdots & \mathbf{0}_{n_{\mathcal{B}_1}} \\ \mathbf{0}_{n_{\mathcal{B}_2}} & \mathbf{1}_{n_{\mathcal{B}_2}} & \cdots & \mathbf{0}_{n_{\mathcal{B}_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_{\mathcal{B}_K}} & \mathbf{0}_{n_{\mathcal{B}_K}} & \cdots & \mathbf{1}_{n_{\mathcal{B}_K}} \end{bmatrix}.$$

Owing to the coupling constraint on $\mathbf{S}_{i,k} = \Pi(\boldsymbol{\mu}_i, \boldsymbol{\nu}_k)$, the summation over elements in $\mathbf{S}_{i,k}$ equals 1 and the graph embedding for $\mathcal{G}_i$ can be further simplified as:

$$\mathbf{w}_i = [\gamma_{i,1}, \cdots, \gamma_{i,K}]. \tag{14}$$

Similarly, for a subgraph $\mathcal{S} \in \mathcal{G}_i$, we can generate the subgraph embedding as $\mathbf{w}_{\mathcal{S}} = \sum_{v_p \in \mathcal{S}} \mathbf{z}_{v_p}\mathbf{W}$.

As stated before, we assume no prior knowledge on nodes and use uniform distributions to represent graphs. Nonetheless, when mass distribution is available or can be learned (Vincent-Cuaz et al., 2022), stronger graph and node embeddings can be further generated.

Graph Reconstruction. The row-normalized optimal coupling $\hat{\mathbf{S}}_{i,k} = n_i\mathbf{S}_{i,k}$ acts as a soft permutation matrix describing the node correspondence between $\mathcal{G}_i$ and $\mathcal{B}_k$. Therefore, we approximate the reconstructed $\mathcal{G}_i$ from $\mathcal{B}_k$, denoted as $\tilde{\mathcal{G}}_{i,k} = \{\tilde{\mathbf{A}}_{i,k}, \tilde{\mathbf{X}}_{i,k}\}$, as the ”realigned” matrix based on $\hat{\mathbf{S}}_{i,k}$ as follows

$$\tilde{\mathbf{A}}_{i,k} = \hat{\mathbf{S}}_{i,k}\mathbf{A}_k\hat{\mathbf{S}}_{i,k}^{\intercal}, \quad \tilde{\mathbf{X}}_{i,k} = \hat{\mathbf{S}}_{i,k}\mathbf{X}_k. \tag{15}$$

Besides, the posterior probability $\gamma_{i,k}$ describes the correspondence between $\mathcal{G}_i$ and $\mathcal{B}_k$. Therefore, the overall reconstructed graph of $\mathcal{G}_i$, denoted as $\tilde{\mathcal{G}}_i = \{\tilde{\mathbf{A}}_i, \tilde{\mathbf{X}}_i\}$, is approximated by the integration of $\tilde{\mathcal{G}}_{i,k}$ based on $\gamma_{i,k}$ as

$$\tilde{\mathbf{A}}_i = \sum_{k=1}^{K} \gamma_{i,k}\tilde{\mathbf{A}}_{i,k}, \quad \tilde{\mathbf{X}}_i = \sum_{k=1}^{K} \gamma_{i,k}\tilde{\mathbf{X}}_{i,k}. \tag{16}$$

It is worth mentioning that the reconstructed graph $\tilde{\mathcal{G}}_i$ belongs to the same metric space of $\mathcal{G}_i$, whereas existing GDL methods (Xu, 2020; Vincent-Cuaz et al., 2022; 2021) can only reconstruct graphs in the atom space. As we will show in the next subsection, the reconstructed graph by Eq. (16) provides a good approximation of the original graph, with an upper bound on the reconstruction error.

3.2. Theoretical Analysis

In this subsection, we provide theoretical analysis regarding the embedding quality, the reconstruction error, the overall convergence and the time complexity. The proofs for all the theorems and propositions are provided in Appendix A.

Bound on Graph Embedding. To elucidate the quality of graph embeddings, we present the following theorem connecting the embedding space with the graph space.

Theorem 3.1. *For two attributed graphs $\mathcal{G}_i, \mathcal{G}_j$, the graph embeddings $\mathbf{w}_i, \mathbf{w}_j$ given by* FRAME *with the 1-*FGW *distance satisfy the following inequality:*

$$\|\mathbf{w}_i - \mathbf{w}_j\|_1 \le \|\log\mathbf{w}_i - \log\mathbf{w}_j\|_1 \le 2K\sigma\mathrm{FGW}_{1,\alpha}(\mathcal{G}_i, \mathcal{G}_j),$$

where K is the number of atoms and σ is the length-scale parameter of the RBF kernel.

Generally speaking, the L_1 norm between the graph embeddings is upper bounded by the 1-FGW distance between the original graphs, indicating that the graph embedding space provides a good proxy to the original graph space. We carry out further experiments in Section 4.4 to demonstrate the close correlation between the graph and embedding spaces.

Upper Bound on Reconstruction Error. To quantify the information loss during the embedding process, we provide the following upper bound on the reconstruction error.

Theorem 3.2. *For an attributed graph $\mathcal{G}_i$ and the reconstructed graph $\tilde{\mathcal{G}}_i$ by Eqs.* (15) *and* (16)*, the 2-*FGW *distance between $\mathcal{G}_i$ and $\tilde{\mathcal{G}}_i$ satisfies the following inequality:*

$$\mathrm{FGW}_{2,\alpha}(\mathcal{G}_i, \tilde{\mathcal{G}}_i) \le \sum_{k=1}^{K} \gamma_{i,k}\left[\alpha\Delta_{\mathrm{GW}_k} + (1-\alpha)\Delta_{\mathrm{W}_k}\right],$$

where $\Delta_{\mathrm{GW}_k} = \|\mathrm{diag}(\boldsymbol{\mu}_i)(\mathbf{A}_i - \tilde{\mathbf{A}}_{i,k})\|_F^2$ and $\Delta_{\mathrm{W}_k} = \|\mathrm{diag}(\boldsymbol{\mu}_i^{\frac{1}{2}})(\mathbf{X}_i - \tilde{\mathbf{X}}_{i,k})\|_F^2$.

To the best of our knowledge, we are the first to derive a reconstruction error bound for GDL. The upper bound suggests that the obtained embedding space retains essential information in the original graph space. Experimental results in Section 4.4 show that FRAME achieves relatively low reconstruction errors.

Convergence Analysis. The EM algorithm is guaranteed to converge under exact maximization (Wu, 1983). For the sake of efficiency, it is preferred to evaluate the maximization step inexactly with a suboptimal solution. As shown in the following theorem, the objective function of FRAME is non-decreasing and converges with inexact maximization.

Theorem 3.3. *The objective function in Eq.* (3) *is non-decreasing and converges along the inexact EM process.*

With the above convergence guarantee, we only need to solve the maximization problem in Eq. (6) inexactly by a few BCD iterations in practice, hence dramatically reducing the running time. Further empirical results on the convergence of FRAME are provided in Appendix C.2.

Time Complexity. Without loss of generality, we assume input graphs share a comparable size that is greater than the shared size of atoms (i.e., $\mathcal{O}(n_{\mathcal{G}}) \gg \mathcal{O}(n_{\mathcal{B}})$ nodes and $\mathcal{O}(m_{\mathcal{G}}) \gg \mathcal{O}(m_{\mathcal{B}})$ edges). We have the following time complexity analysis.

Proposition 3.4. *The overall time complexity of* FRAME *is* $\mathcal{O}(TLNK(m_{\mathcal{G}}n_{\mathcal{B}} + n_{\mathcal{G}}n_{\mathcal{B}}^2))$*, where* T *is the number of EM iterations and* L *is the number of BCD iterations.*

Generally speaking, the time complexity of FRAME is (1) quadratic w.r.t. the node number of atom $n_{\mathcal{B}}$ which is always much smaller than the node number of graph $n_{\mathcal{G}}$, and (2) linear w.r.t. the edge number of input graphs, the number of input graphs and the number of atoms. Sharing the same big-O time complexity, FRAME expands the existing GDL methods' (Xu, 2020; Vincent-Cuaz et al., 2022; 2021) ability to generate nonlinear embedding without increasing the time complexity. In practice, FRAME achieves faster computation as the EM algorithm generally converges with fewer iterations than the gradient descent scheme.

4. Experiment

We conduct extensive experiments to validate and verify our proposed FRAME from the following aspects:

- How to interpret the learned embeddings (Section 4.1)?
- How effective are the learned graph (Section 4.2) and node (Section 4.3) embeddings?
- How well does the learned embedding space represent the original graph space (Section 4.4)?

4.1. Understanding the Learned Embeddings

Experiment Setup. To better understand the relationship between input graphs and learned atoms, we apply FRAME on the synthetic graphs generated by the stochastic block model with {1,2,3,5} blocks. For each category, we generate 50 graphs with node numbers randomly sampled between 10 to 20. Besides, for {1,2,3} blocks, 3 auxiliary graphs per category are generated as the supervision for ss-FRAME. We learn two embedding spaces spanned by 3 and 4 atoms with 12 nodes respectively.

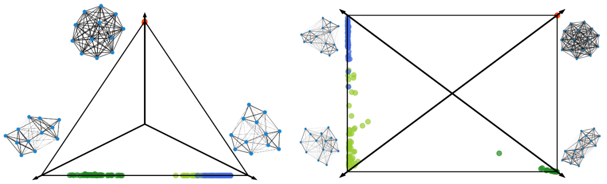

(a) 3-dimensional space (b) 4-dimensional space

Figure 1. Graph embedding spaces for synthetic graphs: ● for 1-block, ● for 2-block, ● for 3-block, and ● for 5-block graphs.

Results and Analysis. The generated graph embedding spaces are shown Fig. 1. The learned graph embeddings are of high quality as the learned atoms recover the block structures of the input graphs and similar graphs are clustered around corresponding atoms. For the 3-dimensional embedding space with less atoms than the graph classes, the 5-block graphs are underrepresented as a mixture of 2-block and 3-block graphs, i.e., partially belonging to both classes. However, when 4 atoms are learned, different graphs are clearly separated in the resulting graph embedding space. Based on this observation, we set the number of atoms to be equal to the number of graph classes in the following experiments.

4.2. Graph Classification and Clustering

Experiment Setup. Three real-world datasets are considered to evaluate the graph embeddings, including ENZYMES (Borgwardt et al., 2005), IMDB-M (Yanardag & Vishwanathan, 2015) and PTC-MR (Kriege & Mutzel, 2012). Detailed dataset description can be found in Appendix D.

Four types of baseline methods are considered, including (1) kernel-based methods: Random Walk (RW) (Kashima et al., 2003), Shortest Path (SP) (Borgwardt & Kriegel, 2005), Weisfeiler-Lehman (WL) (Shervashidze et al., 2011) and Pyramid Match (PM) (Nikolentzos et al., 2017) kernels, (2) embedding-based methods: FGSD (Verma & Zhang, 2017), LDP (Cai & Wang, 2018), MrMine (Du & Tong, 2019) and FEATHER (Rozemberczki & Sarkar, 2020), (3) OT-based methods: FGW (Titouan et al., 2019), WWL (Schulz et al., 2022) and LinearFGW (Nguyen & Tsuda, 2022) and (4) GDL-based methods: GWF (Xu, 2020), GDL (Vincent-Cuaz et al., 2021) and srGW (Vincent-Cuaz et al., 2022).

For a fair comparison with other GDL methods, we fix the number of atoms to be the same as the number of graph

Table 2. Rand Index on graph clustering (%).

Method	ENZYMES	IMDB-M	PTC-MR
RW Kernel	17.6±0.0	34.4±0.0	50.6±0.0
SP Kernel	58.4±0.7	33.6±0.1	50.3±0.0
WL Kernel	61.2±0.9	49.6±0.0	49.9±0.0
PM Kernel	69.0±0.4	52.9±0.0	51.1±0.1
FGSD	61.5±0.0	53.4±0.0	50.4±0.0
LDP	44.5±0.0	40.6±0.0	50.6±0.0
FEATHER	65.0±0.0	50.7±0.0	51.0±0.0
FGW	59.3±0.0	52.1±0.0	49.9±0.0
WWL	24.1±0.0	40.3±5.7	50.6±0.1
linearFGW	57.9±0.5	44.4±0.7	50.9±0.0
GWF	48.8±1.7	**55.4±0.4**	51.5±0.8
GDL	66.5±2.7	53.2±0.3	51.6±0.1
srGW	55.1±0.0	48.1±0.0	50.6±0.0
FRaMe	**69.3±2.3**	54.8±0.5	**52.1±0.4**

Table 3. Accuracy on graph classification (%).

Method	ENZYMES	IMDB-M	PTC-MR
SP Kernel	27.3±5.3	35.0±2.1	55.8±5.8
RW Kernel	20.7±3.6	35.8±3.5	55.8±0.9
WL Kernel	35.3±5.1	49.5±3.2	54.4±9.6
PM Kernel	23.7±5.7	44.1±4.3	52.1±6.9
FGSD	30.5±5.8	38.9±3.4	59.6±7.2
LDP	23.8±3.7	48.7±2.2	56.7±1.9
FEATHER	25.2±4.4	49.3±3.0	55.8±5.9
FGW	26.0±8.3	39.0±7.5	58.2±4.3
WWL	23.8±3.3	33.3±0.3	55.8±0.9
linearFGW	17.0±4.8	32.3±2.7	53.8±4.4
GWF	24.3±5.5	40.3±2.1	58.9±4.4
GDL	37.0±6.7	40.1±2.2	54.4±4.4
srGW	38.3±5.9	45.6±2.2	59.0±6.8
FRaMe	30.2±6.0	40.6±1.9	58.4±4.7
ss-FRaMe	**46.3±4.1**	**51.9±2.7**	**60.3±3.1**

classes (i.e., one representative atom per class), with a fixed atom size $n_{\mathcal{B}} = 5$. For graph classification, we apply 10-fold cross-validation on the benchmark datasets. We use a SVM classifier for classification, and the results are assessed by the classification accuracy. For graph clustering, we apply spectral clustering on top of the graph embeddings, and the results are assessed by the Rand Index (Rand, 1971).

Results and Analysis. The results of graph clustering are reported in Table 2. It is shown that FRaMe achieves an up to 0.5% improvement over the best competitor in Rand Index. We also carried out experiments on graph classification to validate the effectiveness of the proposed semi-supervised FRaMe, and results are shown in Table 3. It is shown that FRaMe achieves comparable performance with other GDL methods. When supervision is incorporated, the semi-supervised FRaMe outperforms all baselines, including the semi-supervised embedding-based methods, with an up to 8.0% improvement.

4.3. Node Clustering

Experiment Setup. We consider four real-world datasets to evaluate the node embeddings, including AIDS (Riesen & Bunke, 2008), ENZYMES (Borgwardt et al., 2005), PROTEINS (Borgwardt et al., 2005) and PTC-MR (Kriege & Mutzel, 2012). Detailed dataset description can be found in Appendix D.

Six well-known node embedding methods are considered, including DeepWalk (Perozzi et al., 2014), GraRep (Cao et al., 2015), node2vec (Grover & Leskovec, 2016), NetMF (Qiu et al., 2018), NodeSketch (Yang et al., 2019) and MrMine (Du & Tong, 2019). We apply spectral clustering on top of the node embeddings and evaluate the clustering results by the Rand Index (Rand, 1971).

Results and Analysis. The results of the node clustering are reported in Table 4. The proposed FRaMe consistently outperforms all the competitors on all datases, achieving an up to 2.5% improvement in the Rand Index compared with the best competitor.

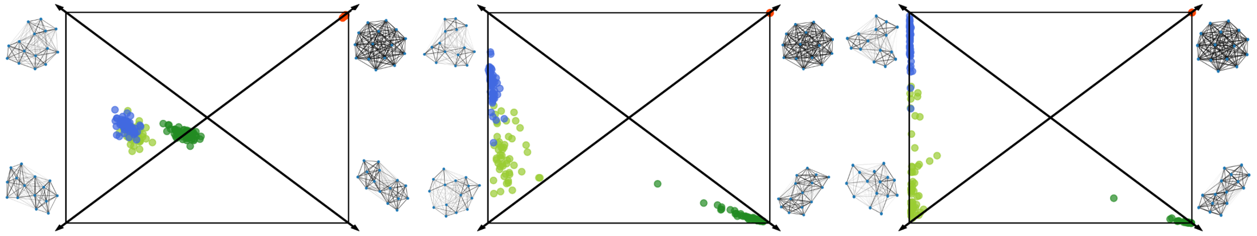
Table 4. Rand Index on node clustering (%).

Method	AIDS	ENZYMES	PROTEINS	PTC-MR
DeepWalk	52.7±0.3	50.1±0.1	50.4±0.1	50.0±0.2
GraRep	47.4±0.0	48.3±0.0	49.4±0.0	50.3±0.0
Node2Vec	43.6±0.1	48.2±0.0	49.5±0.6	50.0±0.2
NetMF	55.4±0.0	50.2±0.0	50.0±0.0	50.2±0.0
NodeSketch	47.9±0.0	48.7±0.0	49.6±0.0	50.7±0.0
MrMine	57.0±0.3	50.6±0.0	51.5±0.0	51.5±0.0
FRaMe	**57.2±0.1**	**53.1±0.1**	**53.5±0.0**	**51.8±0.2**

4.4. Experimental Analysis

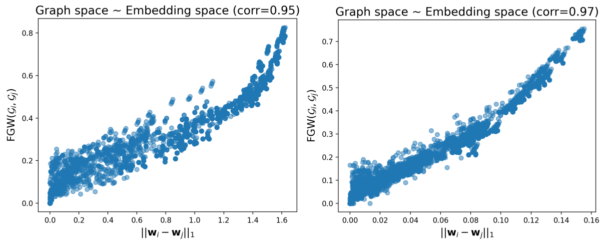
Hyperparameter Study. We analyze the effects of the length-scale parameter σ on the graph embeddings, and the results are shown in Fig. 2. A larger σ generates more deterministic posterior probabilities γ and pushes data samples to the apices of the embedding space, resulting in sparser graph embeddings and clearer block structure in atoms.

The above observation may help guide the model tuning process. When class-specific patterns exhibit in graphs, a large σ is recommended to identify class-specific patterns more deterministically. When common patterns are shared by graphs in different classes, a small σ might be preferred to represent graphs as the mixture of a set of patterns.



(a) σ=10 (b) σ=20 (c) σ=30

Figure 2. Hyperparameter study on σ: ● for 1-block graphs, ● for 2-block graphs, ● for 3-block graphs, and ● for 5-block graphs.



(a) DBLP (b) IMDB-M

Figure 3. Comparison between the graph space and the embedding space. Each point corresponds to a pair of graphs.

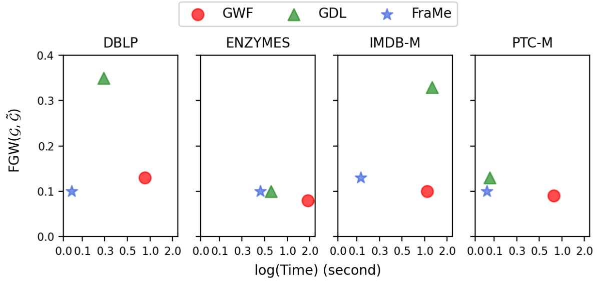


Figure 4. Average graph reconstruction error and running time of different GDL methods.

Graph Embedding Quality. We first evaluate the correlation between the graph and embedding spaces by comparing the FGW distance between graph pairs and the distance between corresponding embeddings. As the results shown in Fig. 3, the learned embedding space well approximates the original graph space with a Pearson coefficient over 0.95.

Besides, we compare the graph reconstruction error measured by the FGW distance between the original graph $\mathcal{G}$ and the reconstructed graph $\tilde{\mathcal{G}}$ of different GDL methods, and results are shown in Fig. 4. For one thing, FRAME achieves the fastest computation thanks to the fast convergence of the EM algorithm. For another, the nonlinear GWF and FRAME achieve smaller reconstruction errors compared with the linear GDL method, validating the necessity of imposing nonlinear graph-atom relationship. Besides, GDL performs relatively poor on two densely-connected datasets DBLP and IMDB-M (refer to data statistics in Table 5), indicating that complicated connectivity patterns may not be well-captured by the linear GDL methods.

5. Related Works

Graph Representation Learning. Extensive efforts have been made in graph representation learning including graph kernel and graph embedding. Graph kernel (Kashima et al., 2003; Borgwardt & Kriegel, 2005; Shervashidze et al., 2011) provides a powerful graph similarity measure by computing the inner product in reproducing kernel Hilbert space (Schölkopf et al., 2002). However, most of the kernel methods require hand-crafted features or predefined rules, resulting in fixed representations that can not be adapted to specific dataset (Vincent-Cuaz et al., 2021). Some graph embedding methods (Perozzi et al., 2014; Grover & Leskovec, 2016) leverage language models to generate representations based on truncated random walks, and some methods (Ribeiro et al., 2017; Donnat et al., 2018) learn embeddings by exploring node structural roles. Graph neural networks (e.g., (Kipf & Welling, 2017; 2016) and its many follow-ups) provide an end-to-end learning framework with promising performance. However, a large amount of labeled data is often required for training and the generated representations are in general not interpretable. Besides, most of the existing work focuses on single graph embedding and may suffer from the embedding space disparity issue when dealing with multiple graphs (Du & Tong, 2019).

Graph Dictionary Learning. Dictionary learning (Mallat, 1999; Mairal et al., 2009; Schmitz et al., 2018) aims to embed data samples into a linear subspace spanned by a set of shared atoms and has achieved great success in vectorial data with wide applications in classification (Raina et al., 2007; Mairal et al., 2009), clustering (Ramirez et al., 2010) and domain adaptation (Ni et al., 2013; Yang et al., 2018). It is of great interest to explore its potential to graph representation learning, but relatively sparse literature exists as graphs lie in disparate metric spaces.

The GDL task is formulated from different aspects. Early works focus on the single-graph setting by factorizing the node attribute matrix with a consistency regularization between embeddings and graph topology (Thanou et al., 2014;

Yankelevsky & Elad, 2016). Recently, GDL is revisited from the reconstructive perspective, which essentially minimizes the Gromov-Wasserstein (GW) discrepancy between the input graphs and corresponding approximations in the atom space. GWF (Xu, 2020) utilizes the GW barycenter as the nonlinear graph approximation, but the resulting bi-level optimization bears a high computational cost. To scale to large graph datasets, GDL (Vincent-Cuaz et al., 2021) approximates input graphs by the linear combination of atoms with an easier-to-solve GW unmixing problem. srGW (Vincent-Cuaz et al., 2022) learns the OT coupling and the target mass distribution simultaneously to avoid pre-defined target mass distribution and further improves the computational efficiency. More recently, RGWD (Liu et al., 2023) proposes a robust GW discrepancy following a minimax formulation, based on which, a GDL method is developed to learn from noisy graph data.

In our work, the GDL task is formulated from the generative perspective by maximizing the likelihood of generating input graphs from the learned atoms. Compared with the existing GDL methods, the generative formulation avoids explicit optimization of the graph embedding (e.g., GW unmixing in (Vincent-Cuaz et al., 2021)) thanks to the closed-loop solution to the expectation step. Besides, by virtue of the fast convergence of EM algorithm, FRAME learns nonlinear embeddings with an empirically faster computational speed. In addition, owing to probabilistic nature underlying the generative approach, multi-level embeddings can be generated with quantitatively identified cross-level relationships. Despite different variants of the GDL formulation, the optimization frameworks essentially follow the BCD optimization scheme by sequentially optimizing the atom (e.g., SGD in GDL and the maximization step in FRAME) and the graph embedding (e.g., GW unmixing in GDL and the expectation step in FRAME).

6. Conclusion

In this paper, we study the graph dictionary learning problem from the generative perspective to learn nonlinear embeddings at multiple levels. An efficient algorithm named FRAME is proposed based on the expectation-maximization algorithm. The proposed FRAME enjoys the same time complexity as the existing GDL with a faster empirical running time. Theoretical analysis shows that the learned embedding space well approximates the original graph space and captures essential graph information with an upper bound on the graph reconstruction error. Extensive experiments on both graph and node-level tasks demonstrate the effectiveness of the proposed FRAME.

Acknowledgements

ZZ, RZ and HT are partially supported by NSF (1947135, 1939725, and 2134079), DARPA (HR001121C0165), NIFA (2020-67021-32799), DHS (17STQAC00001-06-00), and ARO (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 8–pp. IEEE, 2005.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56, 2005.

Cai, C. and Wang, Y. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018.

Cao, S., Lu, W., and Xu, Q. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 891–900, 2015.

Chen, J., Zhu, J., Teh, Y. W., and Zhang, T. Stochastic expectation maximization with variance reduction. *Advances in Neural Information Processing Systems*, 31, 2018.

Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., and Liu, J. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pp. 1542–1553. PMLR, 2020.

Cuturi, M. and Doucet, A. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pp. 685–693. PMLR, 2014.

Donnat, C., Zitnik, M., Hallac, D., and Leskovec, J. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1320–1329, 2018.

Du, B. and Tong, H. Mrmine: Multi-resolution multi-network embedding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 479–488, 2019.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, 2016.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108, 2001.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

Jin, W., Coley, C., Barzilay, R., and Jaakkola, T. Predicting organic reaction outcomes with weisfeiler-lehman network. *Advances in Neural Information Processing Systems*, 30, 2017.

Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In *International Conference on Machine Learning*, pp. 321–328, 2003.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

Kolouri, S., Naderializadeh, N., Rohde, G. K., and Hoffmann, H. Wasserstein embedding for graph learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=AAes_3W-2z`.

Kriege, N. and Mutzel, P. Subgraph matching kernels for attributed graphs. In *International Conference on Machine Learning*, pp. 291–298, 2012.

Ktena, S. I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glocker, B., and Rueckert, D. Distance metric learning using graph convolutional networks: Application to functional brain networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 469–477. Springer, 2017.

Lacoste-Julien, S. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Liu, W., Xie, J., Zhang, C., Yamada, M., Zheng, N., and Qian, H. Robust graph dictionary learning. In *International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=qxRscesArBZ`.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. Online dictionary learning for sparse coding. In *International Conference on Machine Learning*, pp. 689–696, 2009.

Mallat, S. *A wavelet tour of signal processing*. Elsevier, 1999.

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

Maretic, H. P., El Gheche, M., Chierchia, G., and Frossard, P. Fgot: Graph distances based on filters and optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7710–7718, 2022.

McLachlan, G. J. and Rathnayake, S. On the number of components in a gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(5):341–355, 2014.

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. Finite mixture models. *Annual Review of Statistics and Its Application*, 6:355–378, 2019.

Mémoli, F. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11:417–487, 2011.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Nguyen, D. H. and Tsuda, K. On a linear fused gromov-wasserstein distance for graph structured data. *arXiv preprint arXiv:2203.04711*, 2022.

Ni, J., Qiu, Q., and Chellappa, R. Subspace interpolation via dictionary learning for unsupervised domain adaptation.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2):103–134, 2000.

Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Matching node embeddings for graph similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2429–2435, 2017.

Pan, S., Zhu, X., Zhang, C., and Philip, S. Y. Graph stream classification using labeled and unlabeled graphs. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 398–409. IEEE, 2013.

Pearson, K. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

Peyré, G., Cuturi, M., and Solomon, J. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pp. 2664–2672. PMLR, 2016.

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

Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., and Tang, J. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467, 2018.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine learning*, pp. 759–766, 2007.

Ramirez, I., Sprechmann, P., and Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3501–3508. IEEE, 2010.

Rand, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 385–394, 2017.

Riesen, K. and Bunke, H. Iam graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 287–297. Springer, 2008.

Rozemberczki, B. and Sarkar, R. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1325–1334, 2020.

Rozemberczki, B., Kiss, O., and Sarkar, R. Karate club: an api oriented open-source python framework for unsupervised learning on graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 3125–3132, 2020.

Schmitz, M. A., Heitz, M., Bonneel, N., Ngole, F., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

Schölkopf, B., Smola, A. J., Bach, F., et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Schulz, T. H., Horváth, T., Welke, P., and Wrobel, S. A generalized weisfeiler-lehman graph kernel. *Machine Learning*, pp. 1–29, 2022.

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., and Vazirgiannis, M. Grakel: A graph kernel library in python. *The Journal of Machine Learning Research*, 21(1):1993–1997, 2020.

Sturm, K.-T. The space of spaces: curvature bounds and gradient flows on the space of metric measure spaces. *arXiv preprint arXiv:1208.0434*, 2012.

Thanou, D., Shuman, D. I., and Frossard, P. Learning parametric dictionaries for signals on graphs. *IEEE Transactions on Signal Processing*, 62(15):3849–3862, 2014.

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

Togninalli, M., Ghisu, E., Llinares-López, F., Rieck, B., and Borgwardt, K. Wasserstein weisfeiler-lehman graph kernels. *Advances in Neural Information Processing Systems*, 32, 2019.

Verma, S. and Zhang, Z.-L. Hunt for the unique, stable, sparse and fast feature learning on graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

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

Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. Semi-relaxed gromov-wasserstein divergence and applications on graphs. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=RShaMexjc-x`.

Wu, C. J. On the convergence properties of the em algorithm. *The Annals of statistics*, pp. 95–103, 1983.

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

Xu, H. Gromov-wasserstein factorization models for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6478–6485, 2020.

Xu, H., Luo, D., and Carin, L. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in Neural Information Processing Systems*, 32, 2019a.

Xu, H., Luo, D., Zha, H., and Duke, L. C. Gromov-wasserstein learning for graph matching and node embedding. In *International Conference on Machine Learning*, pp. 6932–6941. PMLR, 2019b.

Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Yang, B., Ma, A. J., and Yuen, P. C. Domain-shared group-sparse dictionary learning for unsupervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7453–7460, 2018.

Yang, D., Rosso, P., Li, B., and Cudre-Mauroux, P. Nodesketch: Highly-efficient graph embeddings via recursive sketching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1162–1172, 2019.

Yankelevsky, Y. and Elad, M. Dual graph regularized dictionary learning. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):611–624, 2016.

Zeng, Z., Zhang, S., Xia, Y., and Tong, H. Parrot: Position-aware regularized optimal transport for network alignment. In *Proceedings of the ACM Web Conference 2023*, pp. 372–382, 2023.

A. Proof

Theorem 3.1. *For two attributed graphs $\mathcal{G}_i, \mathcal{G}_j$, the graph embeddings $\mathbf{w}_i, \mathbf{w}_j$ given by* FRAME *with the 1-*FGW *distance satisfy the following inequality:*

$$\|\mathbf{w}_i - \mathbf{w}_j\|_1 \leq \|\log \mathbf{w}_i - \log \mathbf{w}_j\|_1 \leq 2K\sigma \mathrm{FGW}_{1,\alpha}(\mathcal{G}_i, \mathcal{G}_j),$$

where K is the number of atoms and σ is the length-scale parameter of the RBF kernel.

Proof. For simplicity, we use $d_{i,j}$ to denote $\mathrm{FGW}_{1,\alpha}(\mathcal{G}_i, \mathcal{B}_k)$. For the k-th element in $\mathbf{w}_i$ and $\mathbf{w}_j$, i.e., $\gamma_{i,k}$ and $\gamma_{j,k}$, according to Eq. (5), we have:

$$\begin{aligned} |\log \gamma_{i,k} - \log \gamma_{j,k}| &= |\sigma(d_{j,k} - d_{i,k}) + \log \frac{\sum_{m=1}^{K} \pi_m e^{-\sigma d_{j,m}}}{\sum_{m=1}^{K} \pi_m e^{-\sigma d_{i,m}}}| \\ &\leq |\sigma(d_{j,k} - d_{i,k})| + |\log \frac{\sum_{m=1}^{K} \pi_m e^{-\sigma d_{j,m}}}{\sum_{m=1}^{K} \pi_m e^{-\sigma d_{i,m}}}| \end{aligned}.$$

We denote the most ’biased’ barycenter as $\mathcal{B}_q$ satisfying $q = \arg\max_m e^{-\sigma(d_{i,m} - d_{j,m})}$, then we have:

$$\begin{aligned} |\log \gamma_{i,k} - \log \gamma_{j,k}| &\leq |\sigma(d_{j,k} - d_{i,k})| + |\log \frac{\pi_q e^{-\sigma d_{j,q}}}{\pi_q e^{-\sigma d_{i,q}}}| \\ &= |\sigma(d_{j,k} - d_{i,k})| + |\sigma(d_{i,q} - d_{j,q})| \\ &\leq 2\sigma d_{i,j} \end{aligned},$$

where the first inequality is due to $q = \arg\max_m e^{-\sigma(d_{i,m} - d_{j,m})}$ and the second inequality is due to the triangle inequality of 1-FGW distance (Titouan et al., 2019). Besides, as $\gamma_{i,k} \leq 1$, we have $|\gamma_{i,k} - \gamma_{j,k}| \leq |\log \gamma_{i,k} - \log \gamma_{j,k}|$. Therefore, we can bound the L_1 norm of graph embeddings by the corresponding FGW distance as follows:

$$\|\mathbf{w}_i - \mathbf{w}_j\|_1 \leq \|\log \mathbf{w}_i - \log \mathbf{w}_j\|_1 \leq 2K\sigma \mathrm{FGW}_{1,\alpha}(\mathcal{G}_i, \mathcal{G}_j),$$

where K is the number of barycenters and σ is the length-scale parameter of the RBF kernel. □

Theorem 3.2. *For an attributed graph $\mathcal{G}_i$ and the reconstructed graph $\tilde{\mathcal{G}}_i$ by Eqs.* (15) *and* (16)*, the 2-*FGW *distance between $\mathcal{G}_i$ and $\tilde{\mathcal{G}}_i$ satisfies the following inequality:*

$$\mathrm{FGW}_{2,\alpha}(\mathcal{G}_i, \tilde{\mathcal{G}}_i) \leq \sum_{k=1}^{K} \gamma_{i,k} \left[\alpha \Delta_{\mathrm{GW}_k} + (1-\alpha)\Delta_{\mathrm{W}_k}\right],$$

where $\Delta_{\mathrm{GW}_k} = \|\mathrm{diag}(\boldsymbol{\mu}_i)(\mathbf{A}_i - \tilde{\mathbf{A}}_{i,k})\|_F^2$ and $\Delta_{\mathrm{W}_k} = \|\mathrm{diag}(\boldsymbol{\mu}_i^{\frac{1}{2}})(\mathbf{X}_i - \tilde{\mathbf{X}}_{i,k})\|_F^2$.

Proof. For simplicity, we omit the graph index i and use subscripts k, l to index atoms $\mathcal{B}_k, \mathcal{B}_l$. Given $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ and $\tilde{\mathcal{G}} = \{\tilde{\mathbf{A}}, \tilde{\mathbf{X}}\}$, the 2-FGW distance has the following inner product form (Peyré et al., 2016; Vincent-Cuaz et al., 2021):

$$\begin{aligned} &\mathrm{FGW}_{2,\alpha}(\mathcal{G}, \tilde{\mathcal{G}}) = \min_{\mathbf{S} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\mu})} \langle (1-\alpha)\mathbf{M} + \alpha \mathbf{L}, \mathbf{S} \rangle \\ &\text{where} \begin{cases} \mathbf{M} = \mathrm{diag}(\mathbf{X}\mathbf{X}^\intercal)\mathbf{1}^\intercal + \mathbf{1}\mathrm{diag}(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\intercal) - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal \\ \mathbf{L} = \mathbf{A}^2 \boldsymbol{\mu} \mathbf{1}^\intercal + \mathbf{1}\boldsymbol{\mu}^\intercal \tilde{\mathbf{A}}^2 - 2\mathbf{A}\mathbf{S}\tilde{\mathbf{A}} \end{cases} \end{aligned},$$

where $\mathrm{diag}(\mathbf{X})$ of a matrix $\mathbf{X}$ means taking the diagonal value of matrix $\mathbf{X}$ and $\mathrm{diag}(\boldsymbol{\mu})$ of a vector $\boldsymbol{\mu}$ means formulating a diagonal matrix with $\boldsymbol{\mu}$ as the diagonal values. Note that $\mathcal{G}$ and $\tilde{\mathcal{G}}$ have the same number of nodes, hence $\boldsymbol{\mu}_{\mathcal{G}} = \boldsymbol{\mu}_{\tilde{\mathcal{G}}} = \boldsymbol{\mu}$. For simplicity, we use $\mathbf{D}_{\boldsymbol{\mu}}$ to denote $\mathrm{diag}(\boldsymbol{\mu})$.

Since $\mathbf{D}_{\boldsymbol{\mu}} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\mu})$ is an suboptimal solution to $\mathrm{FGW}(\mathcal{G}, \tilde{\mathcal{G}})$, we have:

$$\begin{aligned}\mathrm{FGW}_{2,\alpha}(\mathcal{G}, \tilde{\mathcal{G}}) &\leq \langle (1-\alpha)\mathbf{M} + \alpha\mathbf{L}, \mathbf{D}_{\boldsymbol{\mu}} \rangle \\ &= \underbrace{(1-\alpha)\langle \mathrm{diag}(\mathbf{X}\mathbf{X}^\intercal)\mathbf{1}^\intercal + \mathbf{1}\mathrm{diag}(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\intercal) - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal, \mathbf{D}_{\boldsymbol{\mu}}\rangle}_{\text{Wasserstein distance}} + \underbrace{\alpha\langle \mathbf{A}^2\boldsymbol{\mu}\mathbf{1}^\intercal + \mathbf{1}\boldsymbol{\mu}^\intercal\tilde{\mathbf{A}}^2 - 2\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}, \mathbf{D}_{\boldsymbol{\mu}}\rangle}_{\text{Gromov-Wasserstein distance}}.\end{aligned}$$

We first check the Wasserstein distance term. Owing to the marginal constraint $\mathbf{D}_{\boldsymbol{\mu}}\mathbf{1} = \mathbf{1}\mathbf{D}_{\boldsymbol{\mu}}^\intercal = \boldsymbol{\mu}$ and the trace property $\mathrm{Tr}(\mathbf{X}_1^\intercal\mathbf{D}_{\boldsymbol{\mu}}\mathbf{X}_2\mathbf{D}_{\boldsymbol{\mu}}) = \mathrm{Tr}(\mathbf{X}_1 \odot \mathbf{X}_2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal)$, we have:

$$\begin{aligned}&\langle \mathrm{diag}(\mathbf{X}\mathbf{X}^\intercal)\mathbf{1}^\intercal + \mathbf{1}\mathrm{diag}(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\intercal) - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal, \mathbf{D}_{\boldsymbol{\mu}}\rangle \\ =& \mathrm{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\mathbf{X}\mathbf{X}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} + \mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\ =& \mathrm{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\mathbf{X}\mathbf{X}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} + \sum_{k,l=1}^{K}\gamma_k\gamma_l\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\tilde{\mathbf{X}}_k\tilde{\mathbf{X}}_l^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\ =& \mathrm{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\mathbf{X}\mathbf{X}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} + \sum_{k,l=1}^{K}\gamma_k\gamma_l\tilde{\mathbf{X}}_k \odot \tilde{\mathbf{X}}_l\boldsymbol{\mu}^{\frac{1}{2}}\boldsymbol{\mu}^{\frac{1}{2}^\intercal} - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\ =& \mathrm{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\mathbf{X}\mathbf{X}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} + \sum_{k}^{K}\gamma_k\tilde{\mathbf{X}}_k^2\boldsymbol{\mu}^{\frac{1}{2}}\boldsymbol{\mu}^{\frac{1}{2}^\intercal} - 2\mathbf{X}\tilde{\mathbf{X}}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) + \mathrm{Tr}\left(\left(\underbrace{\sum_{k,l=1}^{K}\gamma_k\gamma_l\tilde{\mathbf{X}}_k \odot \tilde{\mathbf{X}}_l - \sum_{k=1}^{K}\gamma_k\mathbf{X}_k^2}_{\text{non-positive}}\right)\boldsymbol{\mu}^{\frac{1}{2}}\boldsymbol{\mu}^{\frac{1}{2}^\intercal}\right), \\ \leq& \mathrm{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\mathbf{X}\mathbf{X}^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} + \sum_{k=1}^{K}\gamma_k\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}\tilde{\mathbf{X}}_k\tilde{\mathbf{X}}_k^\intercal\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}} - 2\sum_{k=1}^{K}\gamma_k\mathbf{X}\tilde{\mathbf{X}}_k^\intercal\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\ =& \sum_{k=1}^{K}\gamma_k\|\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}(\mathbf{X} - \tilde{\mathbf{X}}_k)\|_F^2\end{aligned} \tag{17}$$

where the non-positivity is owing to the following inequality:

$$\begin{aligned}\sum_{k,l=1}^{K}\gamma_k\gamma_l\tilde{\mathbf{X}}_k(i,j)\tilde{\mathbf{X}}_l(i,j) \leq& \sum_{k,l=1}^{K}\frac{1}{2}\gamma_k\gamma_l\left(\tilde{\mathbf{X}}_k(i,j)^2 + \tilde{\mathbf{X}}_l(i,j)^2\right) \\ =& \frac{1}{2}\left(\sum_{k=1}^{K}\gamma_k\tilde{\mathbf{X}}_k(i,j)^2 + \sum_{l=1}^{K}\gamma_l\tilde{\mathbf{X}}_l(i,j)^2\right). \\ =& \sum_{k=1}^{K}\gamma_k\mathbf{X}_k^2(i,j)\end{aligned}$$

We then check the Gromov-Wasserstein distance term. Similarly, we have:

$$\begin{aligned}
&\langle \mathbf{A}^2\boldsymbol{\mu}\mathbf{1}^\intercal + \mathbf{1}\boldsymbol{\mu}^\intercal\tilde{\mathbf{A}}^2 - 2\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}, \mathbf{D}_{\boldsymbol{\mu}}\rangle \\
=& \operatorname{Tr}\left(\mathbf{A}^2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + \boldsymbol{\mu}\boldsymbol{\mu}^\intercal\tilde{\mathbf{A}}^2 - 2\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\
=& \operatorname{Tr}\left(\mathbf{A}^2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + \sum_{k,l=1}^{K}\gamma_k\gamma_l\tilde{\mathbf{A}}_k \odot \tilde{\mathbf{A}}_l\boldsymbol{\mu}\boldsymbol{\mu}^\intercal - \sum_{k=1}^{K} 2\gamma_k\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}_k\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\
=& \operatorname{Tr}\left(\mathbf{A}^2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal + \sum_{k}^{K}\gamma_k\tilde{\mathbf{A}}_k^2\boldsymbol{\mu}\boldsymbol{\mu}^\intercal - \sum_{k=1}^{K} 2\gamma_k\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}_k\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) + \operatorname{Tr}\left(\left(\underbrace{\sum_{k,l=1}^{K}\gamma_k\gamma_l\tilde{\mathbf{A}}_l \odot \tilde{\mathbf{A}}_k - \sum_{k=1}^{K}\gamma_k\tilde{\mathbf{A}}_k^2}_{\text{non-positive}}\right)\boldsymbol{\mu}\boldsymbol{\mu}^\intercal\right). \qquad (18) \\
\leq& \operatorname{Tr}\left(\mathbf{D}_{\boldsymbol{\mu}}\mathbf{A}\mathbf{A}^\intercal\mathbf{D}_{\boldsymbol{\mu}} + \sum_{k}^{K}\gamma_k\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}_k\tilde{\mathbf{A}}_k^\intercal\mathbf{D}_{\boldsymbol{\mu}} - 2\sum_{k=1}^{K}\gamma_k\mathbf{A}\mathbf{D}_{\boldsymbol{\mu}}\tilde{\mathbf{A}}_k\mathbf{D}_{\boldsymbol{\mu}}^\intercal\right) \\
=& \sum_{k=1}^{K}\gamma_k\|\mathbf{D}_\mu(\mathbf{A} - \tilde{\mathbf{A}}_k)\|_F^2
\end{aligned}$$

Combine Eq. (17) and (18), we prove that

$$\mathrm{FGW}_{2,\alpha}(\mathcal{G}, \tilde{\mathcal{G}}) \leq \sum_{k=1}^{K}\gamma_k\left[\alpha\|\mathbf{D}_\mu(\mathbf{A} - \tilde{\mathbf{A}}_k)\|_F^2 + (1-\alpha)\|\mathbf{D}_{\boldsymbol{\mu}}^{\frac{1}{2}}(\mathbf{X} - \tilde{\mathbf{X}}_k)\|_F^2\right].$$

□

Theorem 3.3. *The objective function in Eq.* (3) *is non-decreasing and converges along the inexact EM process.*

Proof. For simplicity, we denote the set of input graphs as $\mathcal{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_N\}$ and we use superscript (t) to denote parameters in the t-th EM iteration. It is shown that for two set of parameters Θ and $\Theta^{(t)}$, the difference between the two objective functions in Eq. (3) is lower bounded as follows (Wu, 1983):

$$\mathcal{L}(\Theta) - \mathcal{L}(\Theta^{(t)}) \geq \mathcal{Q}(\Theta, \Theta^{(t)}) - \mathcal{Q}(\Theta^{(t)}, \Theta^{(t)})$$

where $\mathcal{Q}(\Theta, \Theta^{(t)}) = \sum_{\mathbf{Z}}\left[p(\mathbf{Z}|\mathcal{G}, \Theta^{(t)})\log p(\mathcal{G}, \mathbf{Z}|\Theta)\right]$ is the maximization objective evaluated with parameters at the t-th EM iteration shown in Eq. (6).

We next show that $\mathcal{Q}(\Theta^{(t)}, \Theta^{(t)}) \leq \mathcal{Q}(\Theta^{(t+1)}, \Theta^{(t)})$ even with inexact maximization, i.e., finding a suboptimal solution to Eq. (6) with a few iterations. We use $\mathcal{Q}_{(l)}^{(t)}$ to denote the value of $\mathcal{Q}(\pi_{(l)}^{(t)}, \mathbf{S}_{(l)}^{(t)}, \mathbf{A}_{(l)}^{(t)}, \mathbf{X}_{(l)}^{(t)}, \gamma^{(t)})$ at the l-th BCD iteration and t-th maximization step.

Since Eq. (7)-(11) provides optimal maximization w.r.t. π, $\mathbf{S}$, $\mathbf{A}$ and $\mathbf{X}$ (Bishop & Nasrabadi, 2006; Xie et al., 2020), the value of $\mathcal{Q}$ is non-decreasing along the BCD iteration, i.e., $\mathcal{Q}_{(l)}^{(t)} \leq \mathcal{Q}_{(l+1)}^{(t)}$. Besides, by setting the initial values for the BCD iteration as $\mathbf{A}_{(0)}^{(t+1)} = \mathbf{A}^{(t)}, \mathbf{X}_{(0)}^{(t+1)} = \mathbf{X}^{(t)}$, we have:

$$\mathcal{Q}(\Theta^{(t)}, \Theta^{(t)}) = \mathcal{Q}_{(0)}^{(t+1)} \leq \mathcal{Q}_{(L)}^{(t+1)} = \mathcal{Q}(\Theta^{(t+1)}, \Theta^{(t)})$$

which means $\mathcal{L}(\Theta^{(t+1)}) - \mathcal{L}(\Theta^{(t)}) \geq 0$ always holds even if $\mathcal{Q}_{(L)}^{(t)}$ is not the exact optimal solution to the maximization problem in Eq. (6). Therefore, we prove that the objective function in Eq. (3) is non-decreasing and converges along the inexact EM process of FRAME. □

Proposition 3.4. *The overall time complexity of* FRAME *is $\mathcal{O}(TLNK(m_{\mathcal{G}}n_{\mathcal{B}} + n_{\mathcal{G}}n_{\mathcal{B}}^2))$, where T is the number of EM iterations and L is the number of BCD iterations.*

Proof. Note that graphs $\mathcal{G}_i$ have sparse adjacency matrix with $m_{\mathcal{G}_i}$ non-zero elements for fast computation. With N attributed input graphs with d node features, K barycenters, T expectation-maximization iterations, L block coordinate descent iterations and M conditional gradient iterations, we have the following analysis.

Calculating $\mathbf{M}_{i,k}^{(t)}$, $|\mathbf{C}_{i,k}(x_1,x_2)-\mathbf{C}_{i,k}(y_1,y_2)|^q$ and $\text{FGW}^{(t)}(\mathcal{G}_i,\mathcal{B}_k)$ in Eq. (1) require $\mathcal{O}(n_{\mathcal{G}}^2 d)$, $\mathcal{O}(m_{\mathcal{G}} n_{\mathcal{B}})$ and $\mathcal{O}(n_{\mathcal{G}} n_{\mathcal{B}})$ complexity. Calculating all the $\gamma_{i,k}^{(t)}$ and $\pi_k^{(t)}$ in Eq. (5) and (7) both require $\mathcal{O}(NK)$ in total. Calculating $\mathbf{S}_{i,k}^{(t)}$ in Eq. (9) requires $\mathcal{O}(M n_{\mathcal{G}} n_{\mathcal{B}})$ complexity. Calculating $\mathbf{A}_k^{(t)}$ and $\mathbf{X}_k^{(t)}$ in Eq. (10) and (11) require $\mathcal{O}(N n_{\mathcal{G}} n_{\mathcal{B}}^2)$ and $\mathcal{O}(N d n_{\mathcal{G}} n_{\mathcal{B}})$ complexity respectively. Ignoring non-trivial terms, the overall complexity of FRAME is $\mathcal{O}(TLNK(m_{\mathcal{G}} n_{\mathcal{B}} + n_{\mathcal{G}} n_{\mathcal{B}}^2))$. □

B. Variants and Discussion

Graph Generation Kernel. The RBF kernel is adopted in FRAME for three reasons. First, the exponential term in Eq. (4) amplifies the gap between graph distances with large σ, resulting in sparse embeddings. Second, the RBF kernel maps the origin graphs into an infinite dimensional space, hence generating highly nonlinear embeddings. Third, with the RBF kernel, the maximization step corresponds to the FGW barycenter problem and can be efficiently solved by the BCD iteration (Titouan et al., 2019). Other exponential kernels such as the Gaussian kernel and the squared exponential kernel can also be adopted with minor modifications. Besides, it is also feasible to adopt other types of non-exponential kernel functions, but the resulting maximization problem may require the computationally more costly gradient descent for optimization.

Reducing Atom Redundancy. It is desirable to generate discriminant atoms capturing diverse graph patterns to reduce redundancy. Therefore, we propose a regularization term that minimizes the mutual likelihood among atoms as follows

$$\Omega(\Theta) = -\beta \sum_{k=1}^{K}\sum_{l=1}^{K} \log p(\mathcal{B}_k|\mathcal{B}_l) \tag{19}$$

And the resulting objective function is formulated as

$$\arg\max_{\pi_k,\mathcal{B}_k} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{i,k}[\log \pi_k - \sigma \text{FGW}(\mathcal{G}_i,\mathcal{B}_k)] + \beta \sum_{\substack{k=1\\l=1}}^{K} \sigma \text{FGW}(\mathcal{B}_k,\mathcal{B}_l)$$

To solve the regularized GDL problem, we can follow the similar EM process but with a modified maximization step. Specifically, we can still adopt Eq. (7) to optimize π_k as the regularization term is decoupled with π_k. To optimize $\mathcal{B}_k$, we follow the BCD iteration where all the variables except $\mathcal{B}_k$ are fixed, and the optimization problem is formulated as

$$\arg\min_{\mathcal{B}_k} \sum_{i=1}^{N} \gamma_{i,k} \text{FGW}(\mathcal{G}_i,\mathcal{B}_k) - \beta \sum_{l=1}^{K} \text{FGW}(\mathcal{B}_l,\mathcal{B}_k)$$

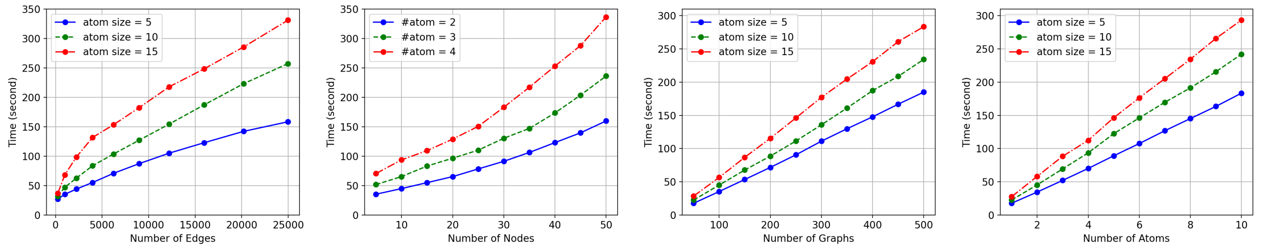
The above problem can be regarded as a modified FGW barycenter problem with negative distances. We can still adopt the CG in Eq. (9) with a modified gradient to optimize $\mathbf{S}_{i,k}$. However, the negative terms would violate the positive semidefinite prerequisite (Peyré et al., 2016) for applying Eq. (10) and destroy the quadratic approximation (Cuturi & Doucet, 2014) in Eq. (11). Therefore, we may not follow the closed-loop solutions in Eqs. (10)-(11), but adopt gradient descent for optimization.

C. Additional Experiments

C.1. Scalability Analysis

We conduct scalability analysis on the synthetic Erdös-Rényi random graphs, and results are shown in Fig. 5.

As shown in Fig. 5(a), the running time is sublinear for small graphs and linear for large graphs w.r.t. the number of graph edges $m_{\mathcal{G}}$. This is mainly because the running time is dominated by the $\mathcal{O}(n_{\mathcal{G}} n_{\mathcal{B}}^2)$ term for small graphs. Besides, in Fig. 5(b), the running time is superlinear w.r.t. the number of atom nodes $n_{\mathcal{B}}$. For Fig. 5(c) and 5(d), the running time is linear w.r.t. the number of graphs N and the number of atoms K, which is consistent with our analysis in Proposition 3.4.

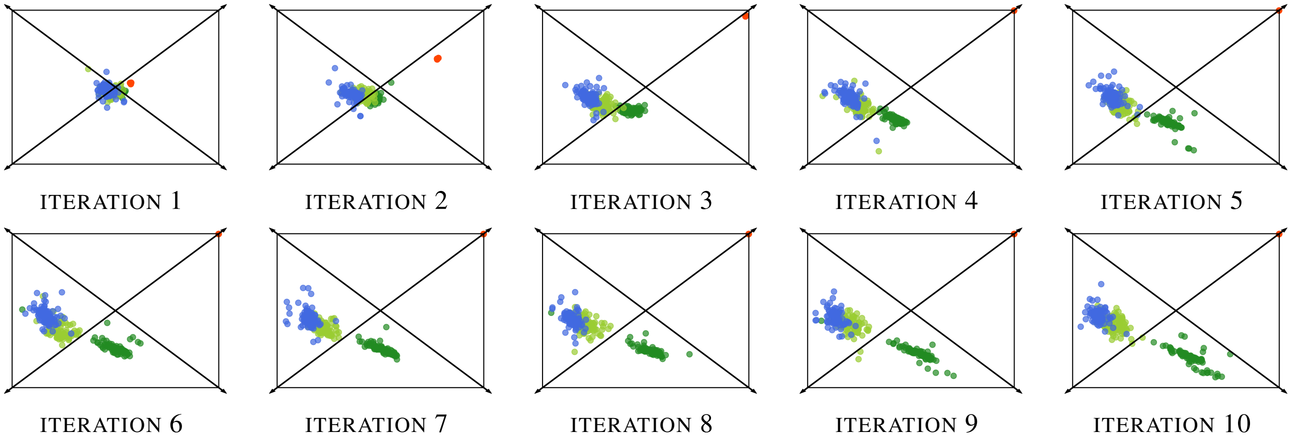


(a) Time - Graph Size (b) Time - Atom Size (c) Time - # Graph (d) Time - # Atoms

Figure 5. Scalability analysis.

C.2. Graph Embedding Space Evolution

We visualize the evolution of the graph embedding space along the EM training process, and the results are shown in Fig. C.2. Along the EM optimization process, the cluster structure gradually appears where similar graphs are closely clustered and dissimilar graphs are pushed far away in the embedding space. Besides, we show how the value of the objective function in Eq. (3) changes along the optimization process in Fig. 7. We observe that the EM algorithm empirically converges after 6 iterations, which validate our claim that FRAME achieves empirically faster computation than the existing GDL methods that adopts the stochastic gradient descent for solution.



ITERATION 1 ITERATION 2 ITERATION 3 ITERATION 4 ITERATION 5

ITERATION 6 ITERATION 7 ITERATION 8 ITERATION 9 ITERATION 10

Figure 6. Evolution of the graph embedding space: ● for 1-block, ● for 2-block, ● for 3-block, and ● for 5-block graphs.

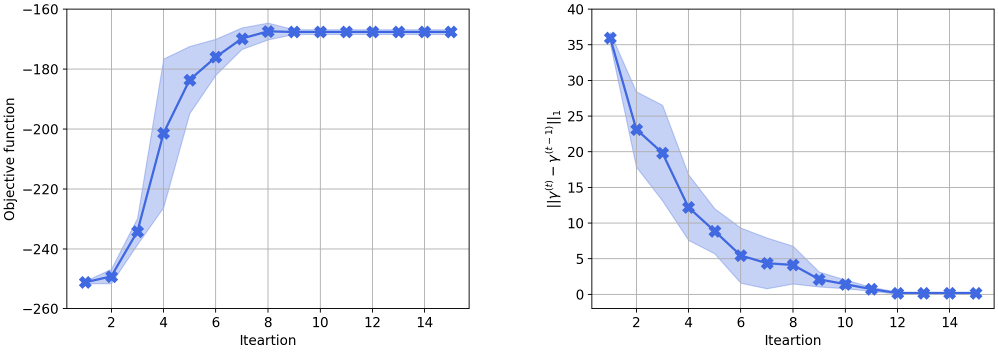
C.3. Convergence Analysis

We empirically evaluate the convergence of the proposed FRAME on the DBLP dataset. Specifically, we assess the convergence by measuring the value of the objective function in Eq. (3) and the difference between two consecutive posterior probability (i.e., $\|\gamma^{(t)} - \gamma^{(t-1)}\|_1$) along the EM iterations. The experiments are repeated 5 times, and we report the mean and standard deviation of the results, which are shown in Fig. 7.

It is shown in Fig. 7(a) that the objective function is non-decreasing along the optimization process and eventually converges to a local optimum after several iterations. Besides, Fig. 7(b) demonstrates that the difference between two consecutive posterior probability $\gamma^{(t-1)}$ and $\gamma^{(t)}$ approaches zero as the optimization proceeds. These results empirically validates our theoretical convergence analysis and provide further evidence supporting our claim that the proposed FRAME based on EM optimization has empirically faster convergence rate than the SGD-based optimization adopted by many GDL methods.

C.4. Learning Multi-Scale Patterns

Capture graph patterns at multiple scales is crucial, but most of the existing GDL methods (Vincent-Cuaz et al., 2021; Liu et al., 2023) fail to achieve this capability as they require atoms to be the same size. In this subsection, we provide



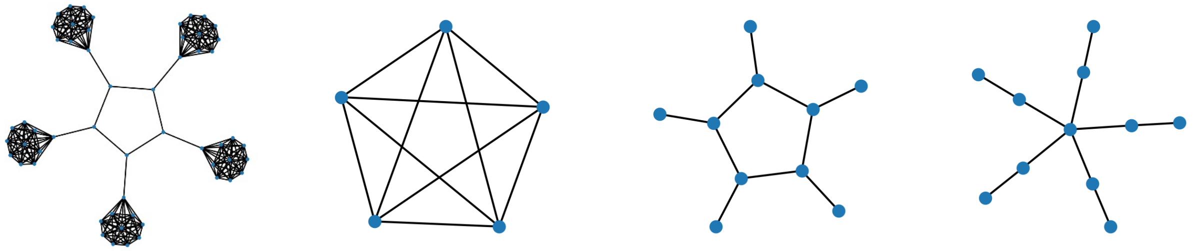
(a) Values of the objective function.

(b) Difference between two consecutive γ.

Figure 7. Convergence analysis of the proposed FRAME.

experimental results to validate FRAME's ability of generating atoms with different sizes to capture multi-scale patterns.

We consider an input graph with patterns at multiple scales shown in Fig. 8(a), which includes a central cycle graph with outreaching edges connecting fully-connected blocks, and apply FRAME to learn 3 atoms with sizes in $\{5, 10, 11\}$. As shown in Fig. 8, FRAME successfully identifies three principal patterns in Figs. 8(b)-8(d): a fully-connected atom, a cycle atom and a star atom.



(a) Input graph

(b) Fully-connected atom

(c) Cycle atom

(d) Star atom

Figure 8. Learning graph patterns at multiple scales.

C.5. Graph Reconstruction

We first analyze how the choice of atoms affects the graph reconstruction process. We consider different atom sizes in $\{5, 10, 15, 20, 25, 30\}$ and different number of atoms in $\{2, 3, 4, 5, 6, 7\}$, and results are shown in Fig. 9.

First, the reconstruction error achieves relatively small values with an atom size that is close to the graph size (i.e., $n_{\mathcal{G}} = 9$ for DBLP, $n_{\mathcal{G}} = 32$ for ENZYMES, $n_{\mathcal{G}} = 13$ for IMDB-M, and $n_{\mathcal{G}} = 14$ for PTC-MR as shown in Table 5). For one thing, graph patterns may not be well captured with small atoms, resulting in high reconstruction errors. For another, when the atom size is big, the reconstruction error either stays unchanged or even increases as the redundancy in atoms may otherwise introduce noises into the reconstruction process.

Second, more atoms lead to smaller reconstruction error. More atoms generate a higher dimensional embedding space, resulting in smaller information loss during the embedding process and more accurate graph reconstruction. Interestingly, the most significant reduction of the reconstruction error happens at the point where the number of atoms equals the number of graph classes. This shows that each atom captures the discriminant pattern in each graph class, and therefore, FRAME can generate an embedding space with little redundancy with properly chosen parameters.

We also provide the visualization of some original graphs and corresponding reconstructed graphs from four datasets, as shown in Fig. C.5. Note that the reconstructed graphs are weighted graphs, and the edge widths are proportional to the corresponding weights. By regarding the edge weights as the probability of connecting two nodes, the proposed FRAME

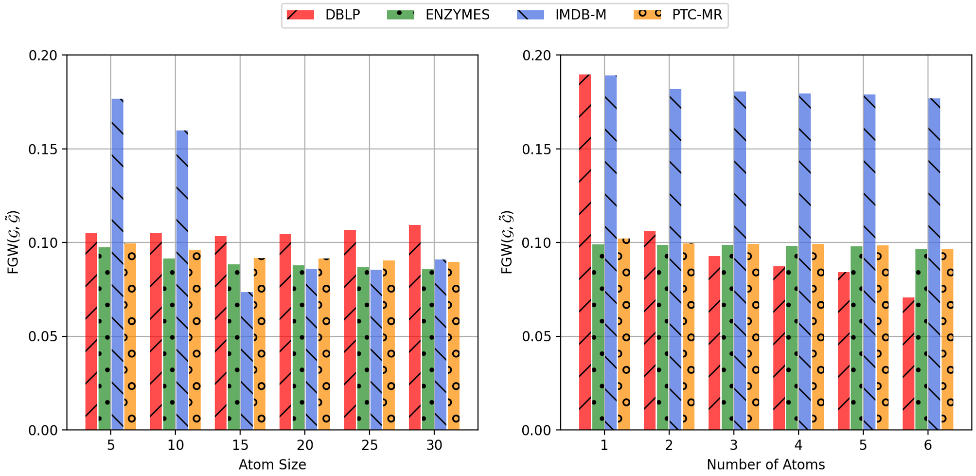


Figure 9. Graph reconstruction error analysis.

can be applied to other tasks such as link prediction and recommendation.

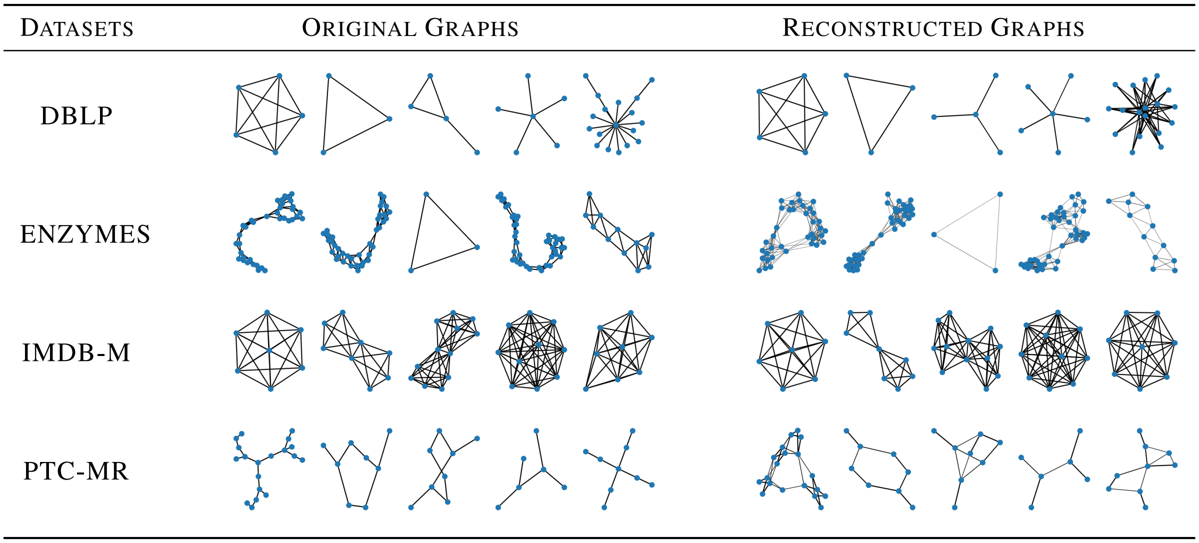


Figure 10. Visualization of graph reconstruction.

D. Reproducibility

Dataset Descriptions. All the real-world datasets we use are from (Morris et al., 2020) and available online[1]. Here we briefly summarize the datasets used in the experiments:

- `AIDS` (Riesen & Bunke, 2008) is a set of chemical graphs, where each graph represents one molecular compound with nodes as atoms and edges as covalent bonds. The binary graph labels indicate whether molecules have activity against HIV or not. The node labels correspond to the atom type.

- `DBLP` (Pan et al., 2013) is a set of biological networks in computer science, where each graph represents one publication with nodes as the paper ID or keywords and edges as citations. The binary graph labels indicate the published conference (DBDM/CVPR) of the paper.

[1] https://chrsmrrs.github.io/datasets/

- ENZYMES (Borgwardt et al., 2005) is a set of enzyme graphs where each graph represents the protein tertiary structure with nodes as atoms and edges as chemical bonds. The graph labels indicate the top-level enzyme classes. The node labels correspond to atom type.
- IMBD-M (Yanardag & Vishwanathan, 2015) is a set of movie collaboration networks, where each graph represents one movie with nodes as actor/actress and edges indicating whether two actors/actresses co-appear in the same movie. The graph labels indicate the movie categories (Comedy/Romance/Sci-Fi).
- PROTEINS (Borgwardt et al., 2005) is a set of proteins where each graph represents the protein structure with nodes as amino acids and edges as chemical bonds. The graph labels indicate whether the proteins are enzymes or non-enzymes. The node labels correspond to atom type and the node attributes represent node chemical features.
- PTC-MR (Helma et al., 2001) is a set of chemical graphs, where each graph represents one molecule with nodes as atoms and edges as chemical bonds. The binary graph labels indicate the carcinogenicity in male rats. The node labels correspond to the atom type.

Table 5. Dataset statistics.

DATASET	#GRAPHS	#NODES	#EDGES	SPARSITY	#FEATURES	#GRAPH CLASS	#NODE CLASS
AIDS	2,000	17.80	18.40	0.06	4	2	38
DBLP	19,456	9.13	19.48	0.23	NONE	2	NONE
ENZYMES	600	31.64	61.85	0.06	18	6	3
IMDB-M	1,500	12.74	53.88	0.33	NONE	3	NONE
PROTEINS	1,113	43.31	77.79	0.04	1	2	3
PTC-MR	344	13.88	14.18	0.07	NONE	2	19

Machine Configuration and Code. The proposed method is implemented in Python based on the POT toolbox (Flamary et al., 2021). The graph kernel methods are based on the *GraKel* library (Siglidis et al., 2020), and the embedding-based methods are based on the *Karate Club* library (Rozemberczki et al., 2020). All experiments are conducted on the Linux platform with an Intel Xeon Gold 6240R CPU and an NVIDIA Tesla V100 SXM2 GPU. The code is implemented by authors from the University of Illinois and available at https://github.com/zhichenz98/FraMe-ICML23.

E. More on Related Works

Optimal Transport on Graphs. The OT theory (Peyré et al., 2019) compares two distributions by finding the optimal coupling minimizing a predefined cost. There has been a recent interest on applying OT on structured data such as graphs with wide applications on graph comparison, graph alignment, graph dictionary learning and so on. The key idea is to represent graphs as distributions and optimize a probabilistic coupling under the Wasserstein discrepancy (Nikolentzos et al., 2017; Maretic et al., 2019; Togninalli et al., 2019) or the Gromov-Wasserstein discrepancy (Mémoli, 2011; Sturm, 2012). Many works (Chen et al., 2020; Xu et al., 2019b; Xu, 2020; Titouan et al., 2019; Zeng et al., 2023) represent graph as a discrete distribution on the product space of graph topology and node attributes, where elements indicate the weight of different nodes, while other works represent graphs as uniform distributions with Laplacian-like covariance matrices (Maretic et al., 2019; 2022). Afterwards, either the distances are utilized for tasks like graph comparison (Xu et al., 2019b; Titouan et al., 2019) and graph dictionary learning (Vincent-Cuaz et al., 2021; 2022; Xu, 2020; Liu et al., 2023), or the OT couplings are leveraged to model node relationships in graph alignment (Xu et al., 2019a;b; Maretic et al., 2019; 2022; Zeng et al., 2023).

Mixture Models. Mixture models are widely used to model unknown distribution shapes in various fields such as bioinformatics, engineering, and imaging (McLachlan et al., 2019). These models offer a semi-parametric approach to represent the unknown distribution as a combination of base distributions. By optimizing the weights and parameters of these base distributions, the model aims to best fit the data distribution. Initially, the method of moments (Pearson, 1894) was employed to fit the unknown distribution by solving a nonic polynomial. Subsequently, the expectation-maximization algorithm emerged as the predominant approach for optimizing mixture models. The key idea follows the principle of maximum likelihood estimation, which iteratively calculates the conditional expectation of the log likelihood and maximizes the complete likelihood. While the mixture model provides a probabilistic clustering of data samples given the base components, the selection of the model order, i.e., number of base components, is essential to the model performance (McLachlan & Rathnayake, 2014; McLachlan et al., 2019). In this paper, we set the model order to be the same as the number of graph classes, as we expect to have one representative base model (i.e., atom) for each graph class.

F. Future Works and Limitations

As discussed in Appendix B, there are several possible directions to explore that can further benefit the current framework, including:

- `Non-exponential graph generation kernels`: When adopting non-exponential graph generation kernels, the resulting formulation is not a FGW barycenter problem, hence may lack of efficient solution. We may need to further explore efficient solutions other than the computationally costly stochastic gradient descent.
- `Redundancy regularization`: When incorporating the redundancy regularization in Eq. (19), the resulting problem can be regarded as a FGW barycenter problem with negative distances, which, however, can not be directly solved by the current algorithm due to the violation of the semi-definite and quadratic properties.
- `Online learning`: When dealing with graphs that come in sequence, it would be beneficial to build up a model that can incrementally learn from new samples (Vincent-Cuaz et al., 2021). Although the current vanilla EM algorithm is not applicable for online learning, the recent advance on stochastic EM algorithm (Chen et al., 2018) may provide a feasible to develop an online generative GDL method.
- `Learnable mass distribution over graphs`: The current follows a common approach in OT-based graph learning frameworks where probability mass is uniformly distributed on nodes. However, the recently proposed srGW (Vincent-Cuaz et al., 2022) provides a semi-relaxed GW discrepancy by relaxing the marginal constraint over the target graph. By relaxing the marginal constraint over the atom, each atom can function differently for different graphs, hence further increase the representation power of the proposed model.
- `Graph generation`: It would be of great interest to explore FRAME's ability of graph generation. In other words, given a graph embedding space learned by FRAME, how to efficiently sample graphs from the model? This future direction not only provides a feasible way to generate graphs obeying the input graph distribution, but also provides a possible way to construct interpretable graph encoder-decoder and graph GAN models.