

Self-Supervised Bidirectional Learning for Graph Matching

Wenqi Guo, Lin Zhang, Shikui Tu, Lei Xu

Department of Computer Science and Engineering, Shanghai Jiao Tong University
{wenqigu,linzhang,tushikui,leixu}@sjtu.edu.cn

Abstract

Deep learning methods have demonstrated promising performance on the NP-hard Graph Matching (GM) problems. However, the state-of-the-art methods usually require the ground-truth labels, which may take extensive human efforts or be impractical to collect. In this paper, we present a robust self-supervised bidirectional learning method (IA-SSGM) to tackle GM in an unsupervised manner. It involves an affinity learning component and a classic GM solver. Specifically, we adopt the Hungarian solver to generate pseudo correspondence labels for the simple probabilistic relaxation of the affinity matrix. In addition, a bidirectional recycling consistency module is proposed to generate pseudo samples by recycling the pseudo correspondence back to permute the input. It imposes a consistency constraint between the pseudo affinity and the original one, which is theoretically supported to help reduce the matching error. Our method further develops a graph contrastive learning jointly with the affinity learning to enhance its robustness against the noise and outliers in real applications. Experiments deliver superior performance over the previous state-of-the-arts on five real-world benchmarks, especially under the more difficult outlier scenarios, demonstrating the effectiveness of our method.

Introduction

Graph matching (GM) aims to find the structural correspondence of nodes in two graphs or multiple graphs by taking both node and edge similarities (affinities) into account (Zanfır and Sminchisescu 2018; Yan et al. 2016; Wang, Yan, and Yang 2019). It has various real-world applications, e.g., protein matching (Krissinel and Henrick 2004; Sharan and Ideker 2006), molecules comparing (Koch, Kriege, and Humbeck 2019), image matching (Wang, Zhou, and Daniilidis 2018), objects tracking, 2D/3D shapes matching (Vento and Foggia 2013), entity alignment (Mao et al. 2021) and model fusion (Liu et al. 2022). Mathematically, for two-graphs with n_1 and n_2 nodes, GM can be formulated as an NP-hard quadratic assignment problem (QAP) (Loiola et al. 2007):

$$\begin{aligned} \max J(\mathbf{Z}) &= \text{vec}(\mathbf{Z})^\top \mathbf{K} \text{vec}(\mathbf{Z}), \\ \text{subject to } \mathbf{Z}\mathbf{1}_{n_2} &= \mathbf{1}_{n_1}, \mathbf{Z}^\top \mathbf{1}_{n_1} \leq \mathbf{1}_{n_2}, \end{aligned} \quad (1)$$

where $\mathbf{Z} \in \mathbb{R}^{n_1 \times n_2}$ is a (partial) permutation matrix encoding the node-to-node correspondence, $\text{vec}(\mathbf{Z})$ is its column-vectorized version, and $\mathbf{1}_n$ is a column vector of length n whose elements all equal to 1. $\mathbf{K} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ is called affinity matrix, and denotes the node-to-node and edge-to-edge similarity in its diagonal and off-diagonal elements, respectively. Since it is intractable to compute a global optimum for the general QAP, traditional methods are usually approximate relaxation algorithms, under a given affinity matrix (Leordeanu, Hebert, and Sukthankar 2009; Gold and Rangarajan 1996; Yu et al. 2018).

Predefined hand-crafted graph attributes and affinity functions may not fit the underlying structure of the real-world matching problem. Advances in machine learning inspire researchers to construct learnable models for graph representation and affinity metrics, so that efficient GM algorithms may be appropriately developed to solve the practical task (Caetano et al. 2009). Recently, with the strong learning ability of deep neural networks, various deep learning methods have been proposed to solve the GM problem in a data-dependent fashion. They have achieved superior performance over those with fixed graph representation and affinity functions on challenging real benchmarks. However, the existing deep learning GM methods usually require a large amount of training data with ground-truth matching solutions. This limits the applications on real-world tasks where the ground-truth correspondences may be difficult and expensive to annotate.

This paper is concerned with developing a self-supervised learning method for the GM problem. It is difficult to perform self-supervised learning on GM without ground-truth labels as it usually involves a non-differentiable discretization optimization step through which the learning gradients cannot pass. The challenge also lies in the node/edge-wise structural graph representation, which has been demonstrated to significantly affect the performance (Fey et al. 2018). There exist only a few attempts in this direction, which either need the empirically hand-drafted initial node/edge features by shape context (Leordeanu, Sukthankar, and Hebert 2012; Zhao, Tu, and Xu 2021) or adopt a classic GM solver with annealing parameters to gradually approach the discrete solution (Wang et al. 2021), which is approximate and time-consuming. Their performances are still not satisfactory. Motivated by the IA-DSM

scheme (Xu 2019a), which was proposed for solving doubly stochastic matrix (DSM) featured combinatorial tasks with complementary benefit between the inbound yAng-mapping for learning and the outbound yIng-mapping for optimization, we develop a self-supervised bidirectional GM learning method (IA-SSGM) to tackle the above challenges. To summarize, we make the following contributions:

- We propose a robust self-supervised bidirectional deep learning method for solving GM. We not only employ the bidirectional learning to generate pseudo correspondence labels for training, but also permute the input by the pseudo correspondence to produce pseudo samples for affinity computation, which transforms the non-differentiable discrete optimization result into a continuous and differentiable feature form.
- We impose a consistency loss on the probabilistic relaxation of the affinity matrix between the original input and the pseudo sample. We provide a theoretical analysis of the consistency loss, which helps further reduce the matching error and strengthens the capacity of our self-supervised learning.
- We introduce a Graph Contrastive Learning (GCL) jointly with the affinity learning, whose inner causality guarantees its success (Mitrovic et al. 2020). We empirically demonstrate that our method outperforms the existing self-supervised GM learning methods, which significantly narrows the gap between unsupervised and supervised learning in efficiency and quality, while avoiding the time-consuming, costly manual labelling of ground-truth correspondences. Our code is available at <https://github.com/CMACH508/IA-SSGM>.

Related Work

Learning of Graph Matching

Early structure-based shallow models sought to learn different affinity weights of nodes and edges for matching, regardless of the node-wise feature representation and structural information, leading to limited the model capacity (Leordeanu, Sukthakar, and Hebert 2012; Cho, Alahari, and Ponce 2013). Recently, a seminal work (Zanfir and Sminchisescu 2018) firstly adopted a Convolutional Neural Network to extract features from images to build the affinity matrix, and used the differentiable spectral method (Leordeanu and Hebert 2005) to obtain the node correspondence. In this line of work, various pipelines were proposed to extract more dedicated features by effectively embedding the structural information into graph node representation via graph neural network (GNN) (Wang, Yan, and Yang 2019; Yu et al. 2019; Fey et al. 2020; Zhao, Tu, and Xu 2021; Jiang et al. 2021). Some works were devoted to constructing the topology in a parametric fashion through structure learning (Zhang and Lee 2019; Yu et al. 2021). Meanwhile, other investigations focused on the decision part. Rolínek *et al.* incorporated the classical GM solver into the end-to-end pipeline (Rolínek et al. 2020). Wang *et al.* scored the assignment solution via embedding on the so-called association graph whose weighted adjacency matrix is the affinity matrix (Wang, Yan, and Yang 2021). However, these methods

need large amounts of node correspondences as ground-truth labels that are usually labour-intensive to annotate, which restricts their applications to real-world problems.

A few works tried to solve the GM problem in an unsupervised manner (Leordeanu, Sukthakar, and Hebert 2012; Wang et al. 2021; Zhao, Tu, and Xu 2021). Particularly, hand-crafted shape context (Belongie, Malik, and Puzicha 2002) was adopted to generate the initial node features to support the subsequent matching algorithm (Leordeanu, Sukthakar, and Hebert 2012; Zhao, Tu, and Xu 2021). Wang *et al.* proposed a dual-branch framework, and graduated assignment was used in one branch to generate pseudo labels as references to the other branch of network predictions (Wang et al. 2021). Due to the limitations of the feed-forward pipeline, their method’s performance is still not satisfactory. Recently, Bai *et al.* (Bai et al. 2021) using Deep Q-Network on the Maximum Common Subgraph Detection task, which requires the isomorphism between two subgraphs and can be thought as exact GM (Yan et al. 2016). In contrast, our method focuses on the maximum affinity rather than zero-distortion, and employs bidirectional learning to use the predicted results to generate pseudo labels, and is able to further minimize the matching error via a consistency loss for the graph embedding.

Self-Supervised Learning for GNNs

Common taxonomies in recent works (Liu et al. 2021; Xie et al. 2022) categorized the GNN-based self-supervised learning methods into contrastive and predictive models. Graph contrastive learning (GCL) employs pair-wise discrimination as their pretext learning tasks (Xie, Xu, and Ji 2022). It trains GNNs to maximize the mutual information between the augmented instances of the same objects through the so-called InfoNCE loss (Oord, Li, and Vinyals 2018). Many investigations designed graph augmentation methods on different tasks at either node-level or graph-level (Zhu et al. 2020; Hassani and Khasahmadi 2020; You et al. 2020). All these works focused on GCL’s performance on graph or node classification, whereas in this paper, we introduce the GCL to work as a auxiliary module to enhance the graph representation learning for solving GM.

Predictive models trained GNNs to predict certain labels for the input data, including graph reconstruction, property prediction, or self-training prediction (Ding et al. 2022; Hwang et al. 2020; Hamilton, Ying, and Leskovec 2017; Wang et al. 2021). Unlike these approaches that obtained the labels from another branch (task), our method generates the pseudo labels from the network predictions. It further recycles them to produce pseudo-data, which are forced to get identical graph embeddings under a bidirectional paradigm.

Method

Problem Definition and Notation

We give mathematical details of GM below. A graph $\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{E}, \mathbf{A})$ consists of a finite set of nodes $\mathcal{V} = \{1, 2, \dots, n\}$, a nodes feature matrix \mathbf{X} , an adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$ and an edges feature matrix \mathbf{E} , where n denotes the number of nodes $|\mathcal{V}|$. Formally, we are given

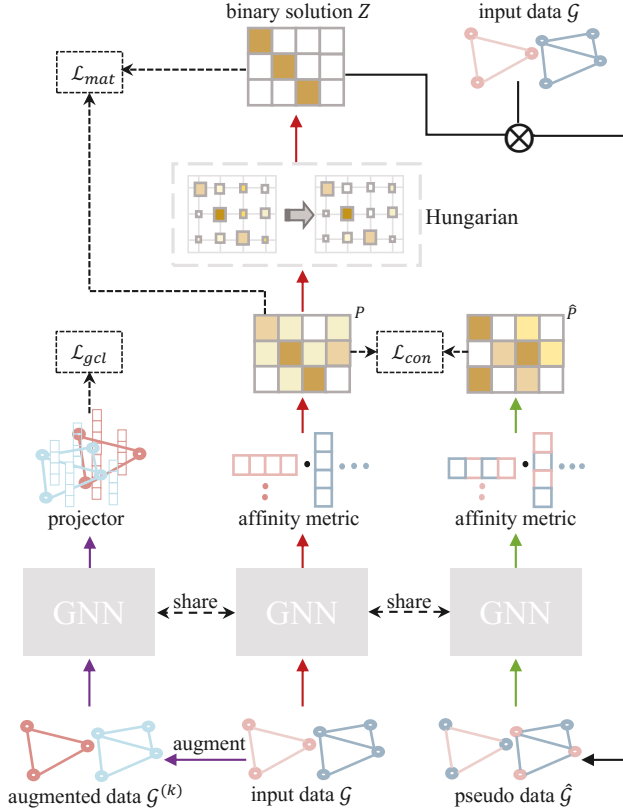


Figure 1: Overview of our self-supervised bidirectional method for GM. The method is built on an affinity learning component for probability and a Hungarian solver for pseudo label generation (the red line). We generate pseudo samples by recycling the pseudo correspondence back to permute the input (the black line), and impose a consistency loss to reduce the matching error to a great extent (the green line). Besides, we introduce a GCL module to enhance the graph representation learning (the purple line).

a source graph $\mathcal{G}_s = (\mathcal{V}_s, \mathbf{X}_s, \mathbf{E}_s, \mathbf{A}_s)$ and a target graph $\mathcal{G}_t = (\mathcal{V}_t, \mathbf{X}_t, \mathbf{E}_t, \mathbf{A}_t)$, with $n_s \leq n_t$, and hope to find a matching matrix $\mathbf{Z} \in \{0, 1\}^{n_s \times n_t}$ to maximize the QAP objective in Eq. (1) subject to the one-to-one mapping constraints $\sum_{j \in \mathcal{V}_t} z_{ij} = 1, \forall i \in \mathcal{V}_s$ and $\sum_{i \in \mathcal{V}_s} z_{ij} \leq 1, \forall j \in \mathcal{V}_t$, where $z_{ij} = 1$ indicates the node i in \mathcal{G}_s is corresponding to the node j in \mathcal{G}_t , and $z_{ij} = 0$ denotes no correspondence. In this sense, \mathbf{Z} deduces an injective mapping from \mathcal{G}_s to \mathcal{G}_t . Recent deep learning GM methods usually learn from data an affinity matrix $\mathbf{K} \in \mathbb{R}^{n_s n_t \times n_s n_t}$ for QAP in Eq. (1) for the subsequent GM solvers. The edge-wise or higher order structural affinity across graphs may also be integrated into a relaxed node-wise affinity matrix $\mathbf{M} \in \mathbb{R}^{n_s \times n_t}$ that can be efficiently solved by linear assignment algorithms. Following the line of learning GM, we consider the scenario of unsupervised learning in the absence of ground-truth correspondences, which are rarely considered in the literature while being critical in real-world

applications.

Overview of Our Method

An overview of our method is given in Fig. 1. It falls into a paradigm of deep bidirectional learning, i.e., an inbound graph representation learning for affinity metrics and an outbound off-the-shelf classic GM solver for pseudo correspondence labels, as well as a feedback loop from the pseudo correspondences to the input space. A GNN encoder is adopted to extract node-wise and structure-wise affinity across graphs. The affinities are normalized as probabilities of correspondences, and the classic Hungarian solver is designated as the reference to generate pseudo correspondence labels for training. The pseudo labels are exploited to guide the affinity learning via a cross-entropy loss. Moreover, we recycle the pseudo GM correspondences to permute the two input graphs to generate pseudo data, which are fed into the GNN encoder again to compute a pseudo estimate for the probability of correspondence. This allows us to devise a bidirectional consistency between the first and the second round (i.e., the pseudo estimate) computation of the probability of correspondence, which is used to guide the self-supervised representation learning. We introduce a GCL module that maximizes the agreement between the embeddings of two different augmented views of the input data, to enhance the graph structure learning by the GNN encoder.

Pseudo Correspondence Generation

For the given input source graph \mathcal{G}_s and target graph \mathcal{G}_t , we first employ a deep shared GNN Ψ_θ to encode them into

$$\mathbf{H}_s = \Psi_\theta(\mathbf{X}_s, \mathbf{A}_s, \mathbf{E}_s), \mathbf{H}_t = \Psi_\theta(\mathbf{X}_t, \mathbf{A}_t, \mathbf{E}_t). \quad (2)$$

We then compute the node-to-node, edge-to-edge, or higher-order affinities into a node-wise affinity matrix \mathbf{M} , similar to (Wang, Yan, and Yang 2019; Fey et al. 2020; Zhao, Tu, and Xu 2021). Therefore, the QAP is turned into a linear assignment problem, i.e., $\max_{\mathbf{Z}} \text{Tr}(\mathbf{Z}^\top \mathbf{M})$, where \mathbf{Z} is constrained to be a permutation matrix with binary entries as in Eq. (1). Specifically, we employ a simple bi-linear mapping to compute \mathbf{M} as

$$\mathbf{M} = \mathbf{H}_s \mathbf{H}_t^\top, \quad (3)$$

where \mathbf{H}_s and \mathbf{H}_t are the embeddings of the two input graphs in Eq. (2).

The permutation matrix variable in the linear assignment problem is further relaxed to a doubly-stochastic matrix, and the resulted problem is a continuous approximation that can be solved by the Sinkhorn algorithm (Sinkhorn 1964; Adams and Zemel 2011). However, Sinkhorn is inefficient to compute due to its alternations between row normalization and column normalization, and has the risk of vanishing gradients (Zhang, Hare, and Prügel-Bennett 2018). Motivated by (Fey et al. 2020; Zhao, Tu, and Xu 2021), we simply apply the row-wise *softmax* normalization on the linear affinity matrix \mathbf{M} in Eq. (3) to generate predictive probability score $\mathbf{P} = [p_{ij}]_{n_s \times n_t}$, which indicates how likely a target node in the column would be matched to each source node in the row. In other words, the binary matching matrix constraints

are relaxed to be $\sum_{j \in \mathcal{V}_t} p_{i,j} = 1, \forall i \in \mathcal{V}_s$. Although the constraint $\sum_{i \in \mathcal{V}_s} p_{i,j} \leq 1, \forall j \in \mathcal{V}_t$ may be violated, this violation is resolved by the subsequent GM solver.

As GM is relaxed to be a linear assignment problem by the affinity learning, we adopt the classic Hungarian algorithm (Kuhn 1955) to compute pseudo correspondence labels, i.e.,

$$\mathbf{Z} = \text{Hungarian}(\mathbf{P}), \quad (4)$$

where \mathbf{P} is the probability score normalized from the affinity matrix \mathbf{M} in Eq. (3), and $\mathbf{Z} = [z_{ij}]_{n_s \times n_t}$ is (partial) permutation matrix with binary entries. Similar to (Zhao, Tu, and Xu 2021), the Hungarian algorithm can be regarded as an improvement operator for the probability score, and the following cross-entropy matching loss is adopted for training the network parameters in affinity learning:

$$\mathcal{L}_{mat} = - \sum_{i \in \mathcal{V}_s, j \in \mathcal{V}_t} (z_{i,j} \log p_{i,j} + (1 - z_{i,j}) \log (1 - p_{i,j})). \quad (5)$$

It should be noted that the above matching loss is induced from the deep bidirectional learning paradigm (Xu 2019b; Zhao, Tu, and Xu 2021). This differs from the dual-branch self-supervised learning framework for GM in (Wang et al. 2021), which is developed based on the common self-supervised paradigm in the literature. The dual-branch is parallelly implemented in a feedforward manner, with one branch for graph representation learning and the other for an off-the-shelf classic GM solver.

Our method is related to the unsupervised learning in (Leordeanu, Sukthakar, and Hebert 2012), which is based on the statistical properties of the affinity matrix \mathbf{M} . A spectral algorithm was derived in (Leordeanu and Hebert 2005) to find a binary solution to the GM problem. Specifically, we restate a theoretical property from (Leordeanu, Sukthakar, and Hebert 2012) as follows:

Proposition 1. *Any normalized vector \mathbf{z} gives a quadratic score that obeys the following optimality bound (Leordeanu, Sukthakar, and Hebert 2012):*

$$\frac{\mathbf{z}^\top \mathbf{K}(\mathbf{w}) \mathbf{z}}{\mathbf{z}_{opt}(\mathbf{w})^\top \mathbf{K}(\mathbf{w}) \mathbf{z}_{opt}(\mathbf{w})} \geq 2 (\mathbf{z}^\top \mathbf{V}(\mathbf{w}))^2 - 1, \quad (6)$$

where $\mathbf{z}_{opt}(\mathbf{w})$ is the optimal solution to the Eq. (1) for a given parameter \mathbf{w} , and $\mathbf{V}(\mathbf{w})$ is the eigenvector of the $\mathbf{K}(\mathbf{w})$.

Accordingly, it immediately follows that maximizing $\mathbf{z}^\top \mathbf{V}(\mathbf{w})$ would maximize the lower bound in Eq. (6) to approach to 1, pushing \mathbf{z} to approximate $\mathbf{z}_{opt}(\mathbf{w})$. We perform the Hungarian algorithm in Eq. (4) to tackle the linear assignment problem. According to (Leordeanu, Sukthakar, and Hebert 2012), it is reasonable to use the probability score \mathbf{P} to replace the eigenvector $\mathbf{V}(\mathbf{w})$ in Eq. (6). Then, the computed \mathbf{Z} in Eq. (4) can be further used by $\max \mathbf{Z}^\top \mathbf{P}$ to train the network parameters. In practice, we use the cross-entropy in Eq. (5), and empirical findings indicate that it is very robust and effective for the final GM performance.

Bidirectional Recycling Consistency

Our bidirectional self-supervised learning paradigm, as illustrated in Fig. 1, enables us to develop a consistency constraint to improve the affinity learning. Based on the generated pseudo correspondence labels \mathbf{Z} by Eq. (4), we further generate pseudo source and target graphs by permuting the original ones via \mathbf{Z} , i.e.,

$$\widehat{\mathbf{X}}_t = \mathbf{Z}^\top \mathbf{X}_s; \quad \widehat{\mathbf{X}}_s = \mathbf{Z} \mathbf{X}_t. \quad (7)$$

We feed the pseudo data $\widehat{\mathcal{G}}_s = (\mathcal{V}_s, \widehat{\mathbf{X}}_s, \mathbf{E}_s, \mathbf{A}_s)$, $\widehat{\mathcal{G}}_t = (\mathcal{V}_t, \widehat{\mathbf{X}}_t, \mathbf{E}_t, \mathbf{A}_t)$ back into the graph encoder to compute the graph embeddings, affinity matrix, and the probability score matrix $\widehat{\mathbf{P}}$, following the same process as Eq. (2)&(3) and the *softmax* normalization layer for probability scores. It is reasonable to enforce an optimality condition that $\widehat{\mathbf{P}}$ should be identical or consistent with \mathbf{P} , which is the probability score matrix for the raw source and target graph, if the current solution \mathbf{Z} is optimal. Mathematically, we compute the consistency by the following cross-entropy loss function:

$$\mathcal{L}_{con} = - \sum_{i \in \mathcal{V}_s, j \in \mathcal{V}_t} \widehat{p}_{i,j} \log p_{i,j}, \quad (8)$$

where $\widehat{p}_{i,j}$ and $p_{i,j}$ are the (i, j) -th element of $\widehat{\mathbf{P}}$ and \mathbf{P} , respectively. When the current solution is far from the optimum, the consistency loss transforms the discrete correspondence error into a continuous feature discrepancy by graph embedding. When the current solution is close to the optimum, the consistency loss still measures the level of sensitivity or smoothness of the graph embedding. Both cases together help reduce the matching error and enhance the graph representation learning. In the following, we provide a theoretical analysis of the consistency loss.

Proposition 2. *Let $\mathcal{G}_s = (\mathbf{X}_s, \mathbf{A}_s)$ and $\mathcal{G}_t = (\mathbf{X}_t, \mathbf{A}_t)$ be the two given graphs, the permutation matrix \mathbf{Z}^* and \mathbf{Z}_θ be the optimal node matching solution and the predictive correspondence with GNN parameter θ , respectively. Define $\mathbf{Z}_\theta^\epsilon = \mathbf{Z}_\theta \mathbf{Z}^{*\top}$, and then $\mathbf{Z}_\theta^\epsilon = \mathbf{I}$ if and only if $\mathbf{Z}_\theta = \mathbf{Z}^*$. For any permutation-equivariant GNN Ψ_θ , we define the following function:*

$$\mathbf{J}(\theta) = \left\| \mathbf{f} \left(\widehat{\mathbf{H}}_s \widehat{\mathbf{H}}_t^\top \right) - \mathbf{f} \left(\mathbf{H}_s \mathbf{H}_t^\top \right) \right\|, \quad (9)$$

where $\widehat{\mathbf{H}}_s = \Psi_\theta(\widehat{\mathcal{G}}_s)$, $\widehat{\mathbf{H}}_t = \Psi_\theta(\widehat{\mathcal{G}}_t)$ are the embedding of synthetic graphs according to Eq. (7)&(2), \mathbf{f} means the row-wise softmax normalization and $\|\cdot\|$ denotes the L_2 norm. Then, the function $\mathbf{J}(\theta)$ achieves its minimum when $\mathbf{Z}_\theta^\epsilon$ becomes the identity matrix \mathbf{I} .

Proof. Based on the definitions, we have

$$\begin{aligned} \mathbf{J}(\theta) &= \left\| \mathbf{f} \left([\Psi_\theta(\mathbf{A}_s \mathbf{Z}_\theta^\epsilon \mathbf{X}_s)] [\Psi_\theta(\mathbf{A}_s \mathbf{Z}_\theta^\epsilon \mathbf{X}_s)]^\top \mathbf{Z}_\theta^{\epsilon 2} \mathbf{Z}^* \right) \right. \\ &\quad \left. - \mathbf{f} \left([\Psi_\theta(\mathbf{A}_s \mathbf{X}_s)] [\Psi_\theta(\mathbf{A}_s \mathbf{X}_s)]^\top \mathbf{Z}^* \right) \right\| \geq 0, \end{aligned}$$

where $[\cdot]$ denotes the output of Ψ_θ in the matrix form, i.e., \mathbf{H} . Then, $\mathbf{J}(\theta) = 0$ when $\mathbf{Z}_\theta^\epsilon = \mathbf{I}$. So minimising $\mathbf{J}(\theta)$ will force \mathbf{Z}_θ to reach \mathbf{Z}^* , which means the matching error is 0. \square

The function $J(\theta)$ actually measures the discrepancy between the predictive probability scores P by the original data and the one \hat{P} by the pseudo data. Specifically, we have $\hat{P} = f(\widehat{H}_s \widehat{H}_t^\top)$, $P = f(H_s H_t^\top)$, and $J(\theta) = \|\hat{P} - P\|$. Proposition 2 indicates that the matching error becomes small when \hat{P} is close to P . Since the cross-entropy loss function is more suitable than L_2 norm for measuring the difference between probability values, in practice, we use Eq. (8) to push \hat{P} towards P , which implicitly reduces the matching error according to the Proposition 2.

From another perspective, we also note that the probability score matrix $f([\Psi_\theta(A_s Z_\theta^\epsilon X_s)] [\Psi_\theta(A_s Z_\theta^\epsilon X_s)]^\top)$ and $f([\Psi_\theta(A_s X_s)] [\Psi_\theta(A_s X_s)]^\top)$ both approach I when optimizing \mathcal{L}_{mat} in Eq. (5). In this way, $J(\theta) \approx \|\mathbf{Z}_\theta^{\epsilon 2} \mathbf{Z}^* - \mathbf{Z}^*\|$, so that minimise $J(\theta)$ will further reduce the error $\mathbf{Z}_\theta^\epsilon$.

Graph Contrastive Learning (GCL) to Enhance Graph Representation Learning

It is critical to perform a high-quality graph representation learning, because the affinity between the two input graphs is computed from the graph embeddings and it may further affect the performance of the subsequent GM solver via Eq. (4). This problem becomes more difficult in the absence of matching labels that contain helpful human experience implicitly. Here, we resort to GCL to enhance the graph representation learning for GM.

We adopt the GCL framework proposed by (You et al. 2020; Zhu et al. 2021), and jointly implement it with affinity learning. For notation brevity, we omit the subscript for source and target graph, because they are treated with the same process in GCL module. Given a batch of N graphs, $\{\mathcal{G}_1, \dots, \mathcal{G}_N\}$, for each graph $\mathcal{G}_i = \{X_i, E_i, A_i\}$, we sample two data augmentation functions $u_1, u_2 \sim \mathcal{U}$ to generate two correlated views $\mathcal{G}_i^{(1)} = u_1(\mathcal{G}_i)$ and $\mathcal{G}_i^{(2)} = u_2(\mathcal{G}_i)$ as a positive pair, where \mathcal{U} is a set of all possible transformation functions, including topology transformation and feature transformation. Then, all of the augmentation views in a batch can be denoted as a set $\{\mathcal{G}_i^{(k)} \mid k \in \{1, 2\}, 1 \leq i \leq N\}$. We then feed each view into the GNN encoder to obtain the augmented node representation: $H_i^{(k)} = \Psi_\theta(\mathcal{G}_i^{(k)})$. They are further transformed by a two-layer multilayer perceptron (MLP) to obtain the final embedding \mathbf{y} for GCL, as below:

$$\mathbf{y}_i^{(k)} = MLP(H_i^{(k)}) / \|MLP(H_i^{(k)})\|. \quad (10)$$

The goal of GCL is to maximize the agreements from the same graph, which in essence can be regarded as a classification problem. We utilize a cross-entropy loss function for each augmented view, i.e.,

$$\mathcal{L}_{gcl} = \sum_{i=1}^N -\log \frac{\psi(\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)})}{\psi(\mathbf{y}_i^{(1)}, \mathbf{y}_i^{(2)}) + \sum_{k, \ell, j \neq i} \psi(\mathbf{y}_i^{(k)}, \mathbf{y}_j^{(\ell)})}, \quad (11)$$

where $\psi(\mathbf{y}, \mathbf{y}') = \exp(\langle \mathbf{y}, \mathbf{y}' \rangle / \tau)$, and τ denotes the temperature parameter, and $\langle \cdot, \cdot \rangle$ means the dot product.

Model Training

The overall loss function for the self-supervised training is obtained by integrating the matching loss in Eq. (5), the bidirectional recycling consistency loss in Eq. (8), and the GCL loss in Eq. (11), i.e.,

$$\mathcal{L}_{overall} = \alpha \mathcal{L}_{mat} + \beta \mathcal{L}_{con} + \gamma \mathcal{L}_{gcl}, \quad (12)$$

where $\alpha, \beta, \gamma \geq 0$ are the hyperparameters to control the relative importance of three loss functions.

It is noted that the consistency loss can be regarded as an extension of the matching loss, by transforming the discrete solution \mathbf{Z} to the continuous probability score \hat{P} . This implies that \mathcal{L}_{mat} and \mathcal{L}_{con} both measure the matching error between the network prediction and the pseudo correspondence. So, it is reasonable to set $\alpha = \beta$. We further set $\alpha = \beta = (1 - \gamma)$ in order to remove the scale redundancy. In practice, we may simply use the GCL module to pre-train the GNN encoder first and fine-tune the network with the matching loss later, by setting γ from 1 to 0 after GCL converges. Although such two-stage training is effective, it is still not optimal. Here, we present a dynamic, annealing technique to adjust the hyperparameters so that the three loss functions are implemented jointly. Specifically, we set

$$\alpha = \beta = \tanh(m/5), \quad \gamma = 1 - \tanh(m/5), \quad (13)$$

where m denotes the learning epoch. When the loss change is smaller than 0.001 among 3 epochs, which means the predictive result of our model converges for each pair graphs, we stop the training.

Experiment

Experimental Settings

We verify our method on datasets including PASCALVOC with Berkeley annotation (Everingham et al. 2010; Bourdev and Malik 2009), WILLOW-OBJECTCLASS (Cho, Alahari, and Ponce 2013), CMU (Belongie, Malik, and Puzicha 2002), CUB2011 (Wah et al. 2011), and IMC-PT-SPARSEGM (Jin et al. 2021; Wang et al. 2021). Per the experiments, we first resize each object in these datasets in bounding box to 256×256 , and interpolate at each keypoint from the pre-trained VGG16's (Simonyan and Zisserman 2014) two feature maps (relu4_2 and relu5_1) via bilinear interpolation, then concatenate them as the graph's node feature. We use the relative Cartesian coordinates of the linked nodes as the edge feature for each graph, which is identical with (Fey et al. 2020). The evaluation metric for all experiments is the average matching accuracy on all classes of each dataset, which is computed between the self-supervised prediction and the ground-truth correspondence label. We conduct comparison experiments with the recent state-of-the-art self-supervised GM methods, i.e., the unsupervised version of IA-GM (Zhao, Tu, and Xu 2021), as well as GANN (Wang et al. 2021).

For fair comparisons, we follow the experimental setup in (Rolínek et al. 2020; Wang et al. 2021) of the keypoints filtering procedure, and include two scenarios:

METHOD	PASCALVOC		WILLOW	CUB2011		CMU	IMC-PT-SPARSEGM	
	INLIER	OUTLIER	INLIER	INLIER	OUTLIER	INLIER	INLIER	OUTLIER
IA-GM	56.1	41.7	93.5	62.0	46.4	98.7	85.3	19.3
GANN	57.2	24.3	92.0	79.0	70.8	100.0	82.3	67.9
IA-SSGM (pretrain)	63.4	61.3	95.6	84.8	84.3	100.0	84.7	71.5
IA-SSGM	65.2	62.8	98.2	86.5	85.6	100.0	86.3	69.0

Table 1: Overall average performance on all datasets under INLIER scenario (same keypoints at both source and target images) and OUTLIER scenario (some key points in target but not in source images).

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	ave
IA-GM	25.1	39.2	41.5	56.6	78.4	79.6	61.4	39.2	33.7	38.5	100.	38.5	39.4	39.6	37.9	98.1	43.1	47.6	93.3	92.1	56.1
GANN	26.6	41.4	44.3	57.9	80.8	79.2	64.8	41.5	34.4	40.1	91.4	40.6	41.7	40.7	40.1	98.4	43.6	50.2	92.9	93.8	57.2
IA-SSGM	38.2	50.3	52.2	61.2	79.0	87.7	69.9	56.7	41.5	55.0	92.8	61.3	65.8	54.9	40.6	98.9	59.3	57.7	89.3	92.8	65.3
IA-GM	18.7	34.8	26.4	35.8	72.8	56.8	37.5	23.9	26.7	26.3	66.8	24.0	28.0	28.1	25.4	92.7	25.5	30.4	70.9	82.6	41.7
GANN	12.6	19.5	16.6	18.5	41.1	32.4	19.3	12.3	24.3	17.2	38.0	12.2	15.9	18.2	19.4	35.5	14.8	15.4	41.5	60.8	24.3
IA-SSGM	33.2	53.2	51.7	66.1	85.0	88.6	77.6	62.3	33.9	49.4	100.	51.4	65.3	46.1	51.6	96.1	51.8	100.	90.3	84.1	62.8

Table 2: Matching accuracies (%) on the PASCALVOC Keypoint dataset. Upper part and bottom part refer INLIER and OUTLIER scenarios, respectively.

- INLIER (keypoints intersection): Only the keypoints present in both source and target image are preserved for the matching task.
- OUTLIER (keypoints inclusion): Target image keypoints have to include all the source image keypoints. The source keypoints that are not present in the target image are then deleted. The target image will contain outliers.

Generally, the OUTLIER scenario is more difficult than the INLIER, and the dataset with a larger number of classes or a higher partial rate is also harder. Since the WILLOW and CMU datasets do not contain any outliers, they are used only in the INLIER scenario.

Our method is implemented in PYTORCH, using PYTORCH GEOMETRIC (Fey and Lenssen 2019) and PYGCL (Zhu et al. 2021) libraries. For all experiments, optimization is done via ADAM (Da 2014) with decaying learning rate. Experiments run on Intel(R) Xeon(R) Gold 6226R CPU (2.90GHz) and one Nvidia A100 (40G) GPU.

Results

We report the overall average matching accuracies in Table 1. It is observed that all methods achieve higher accuracies in INLIER scenario than in OUTLIER. IA-GM and GANN have their own relative advantages under different settings, while our method outperforms them consistently for all cases. Most of the increments by our method are large over both GANN and IA-GM. In particular, our method is extremely robust against others under the OUTLIER scenarios of PASCALVOC, CUB2011, and IMC-PT-SPARSEGM benchmarks, with accuracy increments being 21.1%, 14.8%, and 1.1%, respectively. We also include the GCL-pretrained version of the proposed method for comparisons, i.e., setting γ from 1 to 0 after GCL converges with $\alpha = \beta = 1 - \gamma$ in Eq. (12). This simple hyperparameter setting still makes our method the best in comparisons with IA-GM and GANN.

Notice that all methods perform well on WILLOW and CMU-HOUSE/HOTEL datasets, which are relatively easy with a small number of classes in the data. In the following, we report the detailed results on every benchmark dataset.

Results on Pascal VOC. This dataset (Everingham et al. 2010; Bourdev and Malik 2009) consists of 7020 training images and 1682 test images with 20 classes in total. It contains instances of varying scale, poses and illumination, and the number of keypoints ranges from 1 to 19. We construct graphs via the Delaunay triangulation of keypoints follow with (Wang, Yan, and Yang 2019). We adopt SPLINECNN (Fey et al. 2018) as our GNN encoder with trainable B-spline kernel function conditioned on edge features between node-pairs. The matching accuracies for each of the 20 classes are given in Table 2. It is observed that our method achieves much higher accuracies than the others on over 80% classes, and has comparable results on the rest ones. In particular, our method is extremely robust against the outliers in all kinds of objects. The improvements on both scenarios are significant, with paired t-test p-values being 2.04×10^{-2} and 3.07×10^{-12} , respectively.

Results on CUB2011. The CUB2011 images are from 200 kinds of birds. We randomly sample image pairs from the dataset following (Fey et al. 2020; Wang et al. 2021). As in (Wang et al. 2021), we construct fully-connected graphs, making the graph data more complex. The GNN encoder is implemented by stacking two layers of the GIN operator (Xu et al. 2018). Since there are 200 classes in CUB2011, we report the results for every class in a barplot in Fig. 2 (left). Our method is significantly better than GANN (and IA-GM), according to the paired t-test p-values being 4.73×10^{-15} for INLIER and 5.41×10^{-40} for OUTLIER scenario, respectively.

Results on IMC-PT-SparseGM. This dataset is released by (Wang et al. 2021) based on the IMC-PT 2020 (Jin et al. 2021). Its images are tourist attractions around the world

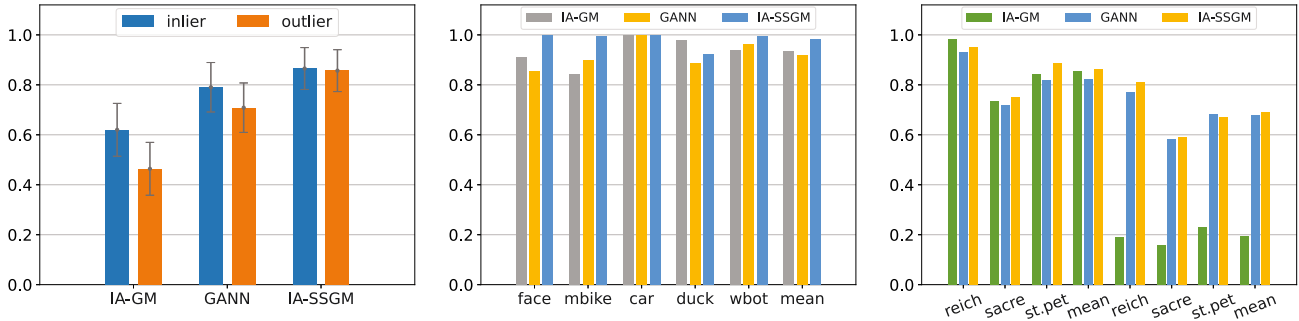


Figure 2: (left) Mean and standard deviation of matching accuracies for all classes in CUB2011 dataset under INLIER and OUTLIER scenarios. (middle) Matching accuracies (%) on the Willow Object dataset. (right) Matching accuracies (%) on the IMC-PT-SparseGM testing dataset. The left four bars refer to INLIER scenarios, and the remaining ones refer to OUTLIER.

METHOD	PASCALVOC		WILLOW	CUB2011		CMU	IMC-PT-SPARSEGM	
	INLIER	OUTLIER	INLIER	INLIER	OUTLIER	INLIER	INLIER	OUTLIER
\mathcal{L}_{mat}	60.9	59.4	94.5	69.9	64.5	100.0	77.1	51.5
$\mathcal{L}_{mat} + \mathcal{L}_{gcl}$	61.1	60.8	96.4	73.5	76.5	100.0	81.3	67.4
$\mathcal{L}_{mat} + \mathcal{L}_{con}$	62.7	60.0	95.8	68.8	56.2	100.0	79.7	64.1
$\mathcal{L}_{mat} + \mathcal{L}_{con} + \mathcal{L}_{gcl}$	65.2	62.8	98.2	86.5	85.6	100.0	86.3	69.0

Table 3: Selectively deactivating loss on PASCALVOC dataset. Average accuracy (%) is reported.

collected from Yahoo Flickr with the number of keypoints ranging from 1 to 50. It is a very challenging dataset because 57.3% of the nodes are invisible. In line with (Wang et al. 2021), we use Reichstag, Sacre Coeur, and St. Peters Square as the testing set and the rest 13 tourism attractions as the training set. Similar to CUB2011, we construct the fully-connected graph. We stack one layer SPLINECNN (Fey et al. 2018) and one layer GIN (Xu et al. 2018) as GNN encoder. The results in Fig. 2 (right) indicate that our method surpasses or at least comparable to the competing methods on all testing classes.

Results on Willow Object and CMU. Since the average matching accuracies on CMU dataset are 100% for most of the methods in Table 1, we omit their detailed accuracies on each category. The WILLOW dataset is collected by (Cho, Alahari, and Ponce 2013) for real images with 5 categories from Caltech256. The images in each category are with relatively fixed pose and their backgrounds are much cleaner than those in PASCALVOC. We construct graphs via the Delaunay triangulation of keypoints. We also use SPLINECNN as the GNN encoder to capture both localized node and global features. Results in Fig. 2 (middle) suggest that our method is very robust on all categories. In particular, the FACE and MBIKE categories are relatively difficult for IA-GM and GANN, and their matching accuracies are improved to be 100% and 99.6% by our method.

Ablation Study

We conduct the ablation study to evaluate the contributions of the pseudo-label induced matching loss in Eq. (4), the pseudo-sample induced bidirectional recycling consistency

in Eq. (8), and the GCL learning module in Eq. (11). Specifically, we activate the designated loss functions in a one-by-one way. We constantly use the matching loss as it is essential for GM matching. The ablation study on all five benchmark datasets is reported in Table 3. We see that the consistency loss and the GCL loss can effectively enhance the matching performance. It is noted that the consistency constraint is able to bring significant performance gain when the GCL module is activated for the baseline.

Conclusion

Learning GM has been made mostly in a supervised manner which requires the ground-truth labels. We propose a robust deep self-supervised bidirectional learning method for GM in the absence of labels. Our method is built on an affinity metric learning component by GNN for probability prediction of correspondence labels, and a classic Hungarian solver for pseudo correspondences that are used to guide the affinity learning. Meanwhile, our method generates pseudo samples by recycling the pseudo correspondences back to permute the raw input graphs, and imposes a consistency constraint between the pseudo sample induced probability prediction and the original prediction, which is theoretically demonstrated to reduce the matching error. Moreover, we employ a GCL module to enhance the graph representation learning against the noise and outliers in real applications. Experiments on five real-world benchmark datasets demonstrate that our method outperforms the current state-of-the-art methods. Admittedly, due to the lack of label, it is more difficult for unsupervised methods to train powerful models, which deserves more work in the future to explore.

Acknowledgements

This work was supported by the National Key R&D Program of China (2018AAA0100700) of the Ministry of Science and Technology of China, BirenTech Research and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). Shikui Tu and Lei Xu are the corresponding authors.

References

- Adams, R. P.; and Zemel, R. S. 2011. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*.
- Bai, Y.; Xu, D.; Sun, Y.; and Wang, W. 2021. Gsearch: Maximum common subgraph detection via learning to search. In *International Conference on Machine Learning*, 588–598. PMLR.
- Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4): 509–522.
- Bourdev, L.; and Malik, J. 2009. Poselets: Body part detectors trained using 3d human pose annotations. In *2009 IEEE 12th International Conference on Computer Vision*, 1365–1372. IEEE.
- Caetano, T. S.; McAuley, J. J.; Cheng, L.; Le, Q. V.; and Smola, A. J. 2009. Learning graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 31(6): 1048–1058.
- Cho, M.; Alahari, K.; and Ponce, J. 2013. Learning graphs to match. In *Proceedings of the IEEE International Conference on Computer Vision*, 25–32.
- Da, K. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Ding, K.; Wang, Y.; Yang, Y.; and Liu, H. 2022. Structural and Semantic Contrastive Learning for Self-supervised Node Representation Learning. *arXiv preprint arXiv:2202.08480*.
- Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.
- Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.
- Fey, M.; Lenssen, J. E.; Morris, C.; Masci, J.; and Kriege, N. M. 2020. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*.
- Fey, M.; Lenssen, J. E.; Weichert, F.; and Müller, H. 2018. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 869–877.
- Gold, S.; and Rangarajan, A. 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 18(4): 377–388.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.
- Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, 4116–4126. PMLR.
- Hwang, D.; Park, J.; Kwon, S.; Kim, K.; Ha, J.-W.; and Kim, H. J. 2020. Self-supervised auxiliary learning with meta-paths for heterogeneous graphs. *Advances in Neural Information Processing Systems*, 33: 10294–10305.
- Jiang, B.; Sun, P.; Zhang, Z.; Tang, J.; and Luo, B. 2021. GAMnet: Robust Feature Matching via Graph Adversarial-Matching Network. In *Proceedings of the 29th ACM International Conference on Multimedia*, 5419–5426.
- Jin, Y.; Mishkin, D.; Mishchuk, A.; Matas, J.; Fua, P.; Yi, K. M.; and Trulls, E. 2021. Image matching across wide baselines: From paper to practice. *International Journal of Computer Vision*, 129(2): 517–547.
- Koch, O.; Kriege, N. M.; and Humbeck, L. 2019. Chemical Similarity and Substructure Searches. In Ranganathan, S.; Gribskov, M.; Nakai, K.; and Schönbach, C., eds., *Encyclopedia of Bioinformatics and Computational Biology - Volume 2*, 640–649. Elsevier.
- Krissinel, E.; and Henrick, K. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, 60(12): 2256–2268.
- Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83–97.
- Leordeanu, M.; and Hebert, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, 1482–1489. IEEE.
- Leordeanu, M.; Hebert, M.; and Sukthankar, R. 2009. An integer projected fixed point method for graph matching and map inference. In *Advances in neural information processing systems*, 1114–1122. Citeseer.
- Leordeanu, M.; Sukthankar, R.; and Hebert, M. 2012. Unsupervised learning for graph matching. *International journal of computer vision*, 96(1): 28–45.
- Liu, C.; Lou, C.; Wang, R.; Xi, A. Y.; Shen, L.; and Yan, J. 2022. Deep Neural Network Fusion via Graph Matching with Applications to Model Ensemble and Federated Learning. In *International Conference on Machine Learning*, 13857–13869. PMLR.
- Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.
- Loiola, E. M.; de Abreu, N. M. M.; Boaventura-Netto, P. O.; Hahn, P.; and Querido, T. 2007. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2): 657–690.
- Mao, X.; Wang, W.; Wu, Y.; and Lan, M. 2021. From alignment to assignment: Frustratingly simple unsupervised entity alignment. *arXiv preprint arXiv:2109.02363*.

- Mitrovic, J.; McWilliams, B.; Walker, J.; Buesing, L.; and Blundell, C. 2020. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*.
- Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Rolínek, M.; Swoboda, P.; Zietlow, D.; Paulus, A.; Musil, V.; and Martius, G. 2020. Deep graph matching via black-box differentiation of combinatorial solvers. In *European Conference on Computer Vision*, 407–424. Springer.
- Sharan, R.; and Ideker, T. 2006. Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4): 427–433.
- Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sinkhorn, R. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2): 876–879.
- Vento, M.; and Foggia, P. 2013. Graph matching techniques for computer vision. In *Image Processing: Concepts, Methodologies, Tools, and Applications*, 381–421. IGI Global.
- Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.
- Wang, Q.; Zhou, X.; and Daniilidis, K. 2018. Multi-image semantic matching by mining consistent features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 685–694.
- Wang, R.; Jiang, S.; Yan, J.; and Yang, X. 2021. Robust Self-supervised Learning of Deep Graph Matching with Mixture of Modes. Submitted to IEEE Transactions of Pattern Analysis and Machine Intelligence.
- Wang, R.; Yan, J.; and Yang, X. 2019. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3056–3065.
- Wang, R.; Yan, J.; and Yang, X. 2021. Neural graph matching network: Learning lawler’s quadratic assignment problem with extension to hypergraph and multiple-graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Xie, Y.; Xu, Z.; and Ji, S. 2022. Self-Supervised Representation Learning via Latent Graph Prediction. *arXiv preprint arXiv:2202.08333*.
- Xie, Y.; Xu, Z.; Zhang, J.; Wang, Z.; and Ji, S. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.
- Xu, L. 2019a. Deep IA-BI and five actions in circling. In *International Conference on Intelligent Science and Big Data Engineering*, 1–21. Springer.
- Xu, L. 2019b. Deep IA-BI and Five Actions in Circling. In *International Conference on Intelligent Science and Big Data Engineering*, 1–21. Springer.
- Yan, J.; Yin, X.-C.; Lin, W.; Deng, C.; Zha, H.; and Yang, X. 2016. A short survey of recent advances in graph matching. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, 167–174.
- 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.
- Yu, T.; Wang, R.; Yan, J.; and Li, B. 2019. Learning deep graph matching with channel-independent embedding and hungarian attention. In *International conference on learning representations*.
- Yu, T.; Wang, R.; Yan, J.; and Li, B. 2021. Deep latent graph matching. In *International Conference on Machine Learning*, 12187–12197. PMLR.
- Yu, T.; Yan, J.; Wang, Y.; Liu, W.; and Li, B. 2018. Generalizing graph matching beyond quadratic assignment model. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 861–871.
- Zanfir, A.; and Sminchisescu, C. 2018. Deep learning of graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2684–2693.
- Zhang, Y.; Hare, J.; and Prügel-Bennett, A. 2018. Learning representations of sets through optimized permutations. *arXiv preprint arXiv:1812.03928*.
- Zhang, Z.; and Lee, W. S. 2019. Deep graphical feature learning for the feature matching problem. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5087–5096.
- Zhao, K.; Tu, S.; and Xu, L. 2021. IA-GM: a deep bidirectional learning method for graph matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3474–3482.
- Zhu, Y.; Xu, Y.; Liu, Q.; and Wu, S. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*.
- Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.