Contrastive Learning for Supervised Graph Matching

Gathika Ratnayaka1 Qing Wang1 Yang Li2

1School of Computing, Australian National University, Canberra, Australia
2School of Information Technology, Deakin University, Melbourne, Australia

Abstract

Deep graph matching techniques have shown promising results in recent years. In this work, we cast deep graph matching as a contrastive learning task and introduce a new objective function for contrastive mapping to exploit the relationships between matches and non-matches. To this end, we develop a hardness attention mechanism to select negative samples which captures the relatedness and informativeness of positive and negative samples. Further, we propose a novel deep graph matching framework, Stable Graph Matching (StableGM), which incorporates Sinkhorn ranking into a stable marriage algorithm to efficiently compute one-to-one node correspondences between graphs. We prove that the proposed objective function for contrastive matching is both positive and negative informative, offering theoretical guarantees to achieve dual-optimality in graph matching. We empirically verify the effectiveness of our proposed approach by conducting experiments on standard graph matching benchmarks.

1 INTRODUCTION

Given two graphs, graph matching aims to establish node correspondences between the graphs. It is fundamental to many real-world applications, such as computer vision [Vento and Foggia 2013], bio-informatics [Sharan and Ideker 2006], and social networks [Zhang et al. 2019]. However, due to the intriguing combinatorial nature of graphs, graph matching has long been known to be an NP-hard problem [Cordella et al. 2004].

Early studies [Cho et al. 2010, Zaslavskiy et al. 2008] on graph matching primarily relied on handcrafting affinities of nodes and edges to leverage structural information from graphs. Graph matching was then formulated as a quadratic assignment problem (QAP) and solved by applying combinatorial optimization techniques [Lawler 1963, Lioiola et al. 2007]. As a result, traditional graph matching methods often lack flexibility and generalizability, failing to capture complex interactions in real world applications. Recently, a number of deep graph matching methods have been proposed to address the limitations of traditional graph matching methods and they have achieved state-of-the-art performance [Wang et al. 2019, Yu et al. 2020, Rolínek et al. 2020, Gao et al. 2021, Wang et al. 2021].

The main idea behind deep graph matching methods is to build an end-to-end learning model which can extract affinities from graphs via differentiable optimisation for finding “soft” node correspondences between graphs. This process often involves a spectral relaxation or a doubly-stochastic relaxation [Zanfir and Sminchisescu 2018, Wang et al. 2020b]. The learning objectives are usually formulated as a cross entropy loss to maximize affinities between pairs of matched nodes as per the ground truth while minimizing affinities between nodes that do not match as per the ground truth, e.g. permutation loss [Wang et al. 2019, Yu et al. 2020] and binary cross entropy loss [Wang et al. 2021]. However, these methods still suffer from undesired behaviours, such as class imbalance, overfitting, and numerical instability [Yu et al. 2020, Gao et al. 2021]. We observe that these issues are largely due to the inherent limitations of cross-entropy loss, which penalize classification errors based on an element-wise comparison between a predicted matching matrix and a ground truth matrix, while ignoring the relationships among matches and non-matches.

On the other hand, contrastive loss has demonstrated superior performance in various self-supervised learning tasks [Wang and Isola 2020, Wang and Liu 2021]. In a contrastive learning setting, given an anchor, there is one positive sample and multiple negative samples. It has been shown that the quality of negative samples has a significant impact on learning [Xu et al. 2022]. Moreover, it has been shown that contrastive learning with explicit hard negative sampling, which penalizes hard negatives while ignoring
easy (uninformative) negative samples during the training, often yields better results [Wang and Liu 2021].

However, we observe that, not only the informativeness of negative samples but also the informativeness of positive samples play an important role in contrastive learning. Furthermore, for graph matching, the one-to-one mapping constraint is crucial, which poses the implication that for each node in a source graph there exists one true match and the others are all non-matches. Thus, unlike existing methods which either removed this constraint from their relaxation or only implicitly considered it in their learning, we explore the power of contrastive loss to effectively incorporate the one-to-one mapping constraint into the learning objective through the choice of negative samples that are most informative with respect to each positive sample.

These observations motivate us to leverage contrastive learning in a supervised graph matching setting. In our work, if two nodes are matched with each other as per the ground truth, such a pair of nodes is considered as a positive sample and a pair of nodes that should not be matched with each other as per the ground truth is considered a negative sample. Within our framework, we propose an explicit hard negative sampling strategy called "hardness attention" to identify hard negative samples, i.e., negative samples that have a high chance of being falsely identified as a positive sample, and propose a new contrastive loss for graph matching. We discuss the theoretical limitations that may occur when adapting the standard contrastive loss from the self-supervised literature [Wang and Liu 2021] to a supervised graph matching setting, and describe how the proposed contrastive loss function possesses theoretical properties that are suitable for graph matching. Moreover, we extend the stable marriage algorithm [McVitie and Wilson 1970; Gale and Shapley 1962] to graph matching for computing the final one-to-one correspondence between two graphs.

Contributions This work perceives supervised graph matching as a contrastive learning task. The main technical contributions of this work are summarized as follows.

- We design a novel hard negative sampling strategy that is well suited for the supervised graph matching setting. This hard negative sampling strategy captures the relatedness and informativeness of positive and negative samples with respect to the corresponding true-matching node pairs.
- We propose a new contrastive loss function for supervised graph matching and theoretically analyze the properties of the proposed loss function. We show that the proposed contrastive loss function is both positive and negative informative.
- We develop a new deep graph matching architecture, called StableGM, which incorporates two key design choices: (1) The Sinkhorn algorithm is used to learn a preference ranking matrix in which each node is associated with a preference order of nodes for matching. (2) We extend the stable marriage algorithm to graph matching which takes a Sinkhorn ranking matrix as input and generates a matching under the one-to-one mapping constraint. We prove that matchings generated by this algorithm are stable.
- Theoretically, we prove that StableGM guarantees the dual-optimality of matching solutions, i.e., nodes in both source and target graphs can achieve an optimal matching simultaneously based on Sinkhorn ranking. Empirically, we conduct experiments on real-world datasets to verify the effectiveness of the proposed graph matching framework.

2 BACKGROUND

We represent a graph as \( G = (V, E, A, X) \), where \( V = \{1, \ldots, |V|\} \) is the set of vertices, \( E \) is the set of edges, \( A \in \mathbb{R}^{V \times V} \) is the adjacency matrix, and \( X \in \mathbb{R}^{V \times d} \) is a node feature matrix with feature vectors of dimension \( d \). Given two graphs, a source graph \( G_S = (V_S, E_S, A_S, X_S) \) and a target graph \( G_T = (V_T, E_T, A_T, X_T) \), let \( |V_S| = m \) and \( |V_T| = n \), and w.l.o.g. \( m \leq n \).

Generally speaking, the graph matching problem is to find a node correspondence matrix \( M \in \{0, 1\}^{m \times n} \) between \( G_S \) and \( G_T \) that satisfies the one-to-one mapping constraint \( \sum_{j=1}^{n} M_{i,j} = 1 \forall i \in V_S \) and \( \sum_{i=1}^{m} M_{i,j} \leq 1 \forall j \in V_T \) and is optimal w.r.t. an objective function \( f(\cdot) \):

\[
\arg \min_{M} f(M; G_S, G_T). \tag{1}
\]

In the supervised learning setting, \( f(\cdot) \) is usually defined in terms of a loss function \( L(\cdot) \). Given two graphs \( G_S \) and \( G_T \), and the ground truth node correspondence matrix \( M^* \), a graph matching model is learnt to minimize \( L(\cdot) \):

\[
\arg \min_{M} L(M^*, M; G_S, G_T). \tag{2}
\]

To make the model differentiable, \( M \) is usually a "soft" matching matrix, such as a doubly stochastic matrix [Fogel et al. 2013].

Problem formulation Let \( \mathcal{G} \) denote a set of graphs on a finite number of nodes and \( \mathcal{M} \) be the set of \( m \times n \) matching matrices, i.e., \( \mathcal{M} = \{ M \in \{0, 1\}^{m \times n} \mid \sum_{j=1}^{n} M_{i,j} = 1 \forall i \in V_S \text{ and } \sum_{i=1}^{m} M_{i,j} \leq 1 \forall j \in V_T \} \). In this work, our goal is to develop a deep graph matching model \( f_{\theta} : \mathcal{G} \times \mathcal{G} \to \mathcal{M} \) by end-to-end training through optimizing a contrastive matching objective.
3 STABLE GRAPH MATCHING

In this section we introduce a novel deep graph matching architecture, Stable Graph Matching (StableGM).

Overview of StableGM Let \( \mathbb{C} \subseteq \mathbb{R}^{m \times n} \) be a set of \( m \times n \) cross-graph node-to-node affinity matrices, \( \mathbb{S} \) be a set of \( m \times n \) rectangular doubly stochastic matrices s.t. \( \mathbb{S} = \{ S \in [0, 1]^{m \times n} \mid S1_m = 1_m, S^T1_m \leq 1_n \} \).

StableGM consists of three main components.

1. **Graph affinity encoding**: a permutation-equivariant, differentiable function \( \psi_{\text{encode}} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{C} \) learns to encode features of two input graphs into a cross-graph node-to-node affinity matrix;

2. **Sinkhorn ranking**: a permutation-invariant, differentiable function \( \psi_{\text{rank}} : \mathbb{C} \rightarrow \mathbb{S} \) produces a preference ranking matrix for nodes in the two input graphs, i.e., for each node in one graph, ranking the preference orders in terms of nodes in the other graph;

3. **Stable matching**: a matching algorithm \( \psi_{\text{match}} : \mathbb{S} \rightarrow \mathbb{M} \) produces a matching matrix (node-to-node correspondences) between two input graphs under the one-to-one mapping constraint.

3.1 GRAPH AFFINITY ENCODING

Graph affinity encoding involves two steps. First, we apply a neural network to encode geometric affinities between nodes, leveraging their features and structural information, into node embeddings. Then, a cross-graph node-to-node affinity layer takes the node embedding matrices of two graphs and constructs a matrix representing affinities between nodes in \( \mathcal{G}_S \) and \( \mathcal{G}_T \).

Concretely, a neural network \( \text{NN}_\theta : \mathbb{G} \rightarrow \mathbb{R}^{[V] \times d} \), parameterized by \( \theta \), takes a graph \( \mathcal{G} \in \mathbb{G} \) as input and returns its node embeddings \( \text{NN}_\theta(\mathcal{G}) \in \mathbb{R}^{[V] \times d} \). Given two graphs \( \mathcal{G}_S \) and \( \mathcal{G}_T \), we apply \( \text{NN}_\theta \) to obtain the node embedding matrices of \( \mathcal{G}_S \) and \( \mathcal{G}_T \), respectively:

\[
\text{NN}_\theta(\mathcal{G}_S) = H^S \in \mathbb{R}^{m \times d} ; \text{NN}_\theta(\mathcal{G}_T) = H^T \in \mathbb{R}^{n \times d}.
\]

Then, the cross-graph node-to-node affinity matrix \( \mathbb{C} \in \mathbb{R}^{m \times n} \) between \( \mathcal{G}_S \) and \( \mathcal{G}_T \) is derived s.t. \( \mathbb{C}_{i,j} \) denotes the affinity between the embedding of node \( i \) in \( \mathcal{G}_S \) and the embedding of node \( j \) in \( \mathcal{G}_T \).

\[
\mathbb{C} = f_{\gamma}(H^S, H^T)
\]

Here, \( f_{\gamma} \) denotes an affinity layer parameterized by \( \gamma \) that encodes the cross-graph node-to-node affinities between \( \mathcal{G}_S \) and \( \mathcal{G}_T \). Different techniques such as the inner product \( \text{Fey} \text{ et al. [2020]} \), and the weighted inner product \( \text{Rolínek et al. [2020]} \) can be used within an affinity layer to calculate the cross-graph node-to-node affinities.
### 3.2 Sinkhorn Ranking

Given a graph affinity matrix that captures node-to-node affinity scores between two graphs, we apply Sinkhorn normalization \cite{Sinkhorn1964} to compute the ranking of nodes in terms of their preference orders for matching.

Following \cite{Wang2020b}, a nonnegative matrix $S^0$ is first obtained via $S^0_{i,j} = \exp(C_{i,j}/\alpha)$, where $\alpha > 0$ is a parameter. In the case where the target graph is larger, i.e., $m < n$, the matrix $S^0$ is padded into a square matrix with small positive values (e.g., $1^{-10}$). Then, the rows and columns of $S^0$ are iteratively normalized into a rectangular doubly-stochastic matrix using an extended Sinkhorn algorithm \cite{Cour2007} [Fey2020]. Specifically, let $\eta(\cdot)$ be the Sinkhorn operator, $\otimes$ be the elementwise Hadamard division, and $1_n$ be a column vector of ones. Starting with $\eta^0(\hat{S}) = S^0$, we have

$$\eta^i(\hat{S}) = f_{row}(f_{col}(\eta^{i-1}(\hat{S}))),$$

where

$$f_{row}(\hat{S}) = \hat{S} \otimes (S1_n1_n^T)$$

$$f_{col}(\hat{S}) = \hat{S} \otimes (1_n1_n^T\hat{S}).$$

Padded elements are discarded from the final output to obtain a rectangular doubly stochastic matrix $S \in \mathbb{S}$. It should be noted that $S_{i,j}$ can be viewed as representing the normalized affinity between $i \in V_S$ and $j \in V_T$.

Within our StableGM framework, any node $i \in V_S$ has a preference order, in which all the nodes in $V_T$ are ranked based on their normalized affinities. Specifically, given any two nodes $j, k$ in $V_T$, if $S_{i,j} > S_{i,k}$, $j$ is ranked higher than $k$ in $i$'s preference order. If $S_{i,j} = S_{i,k}$, the tie is broken arbitrarily to get a strict ranking order (either $j$ can be ranked higher to $k$ or $k$ can be ranked higher to $j$). Any given node $j \in V_T$ also has a preference order over all the nodes in $V_S$.

Given any two nodes $i, k$ in $V_S$, if $S_{i,j} > S_{k,j}$, $i$ is ranked higher than $k$ in $j$'s preference order. If $S_{i,j} = S_{k,j}$, the tie is broken arbitrarily. As the preference orders of the nodes are derived based on $S$, we term $S$ as a Sinkhorn ranking.

### 3.3 Stable Matching Algorithm

Based on the Sinkhorn ranking that yields preference orders of nodes for matching, we leverage the known results on the stable marriage problem \cite{Gale1962} [McVitie1970] to efficiently generate graph matching. We begin with defining the notion of blocking pair.

**Definition 1** (Blocking pair). Given a Sinkhorn ranking $S$ and a matching $M$, a node pair $(i, j)$ with $i \in V_S$ and $j \in V_T$ is a blocking pair w.r.t. $M$ if the following conditions are all satisfied:

- $M_{i,j} = 0$;
- $\exists k \in V_S \setminus \{i\} (M_{k,j} = 1 \land S_{i,j} > S_{k,j})$;
- $\exists k' \in V_T \setminus \{j\} (M_{i,k'} = 1 \land S_{i,j} > S_{i,k'}) \land \forall k'' \in V_T \setminus \{j\} M_{i,k''} = 0$.

Two non-matched nodes $i \in V_S$ and $j \in V_T$ become a blocking pair if $i$ has a higher rank with $j$ than the node that is matched with $j$, and $j$ has a higher rank with $i$ than the node (if any) that is matched with $i$. If there is at least one blocking pair w.r.t. $M$, $M$ is considered as being unstable.

**Definition 2** (Stable matching). A matching $M$ is stable w.r.t. a Sinkhorn ranking $S$ iff $M$ has no blocking pair.

Algorithm \[\] describes our stable matching algorithm, which extends the Male Optimal Stable Marriage (MOSM) algorithm proposed in \cite{McVitie1970} to the graph matching setting. Taking a Sinkhorn ranking $S$ as input, $V_S$ and $V_T$ are assumed to correspond to the male and female parties in the stable marriage problem, respectively. Then, the stable matching algorithm returns a predicted matching matrix $M \in \{0, 1\}^{m \times n}$ that contains one-to-one node correspondences from $V_S$ to $V_T$. Here, $M_{i,j} = 1$ means node $i \in V_S$ is predicted to match with $j \in V_T$ and vice versa.

The following lemma and theorem can be obtained. The proofs are available in the supplementary material.

**Lemma 1.** Given a Sinkhorn ranking $S$, Algorithm \[\] can always return a matching $M$ that satisfies the one-to-one mapping constraint: $\sum_{j=1}^{n} M_{i,j} = 1 \forall i \in V_S$ and $\sum_{i=1}^{m} M_{i,j} \leq 1 \forall j \in V_T$.

**Theorem 1.** Let $M$ be a matching returned by Algorithm \[\] over a Sinkhorn ranking $S$. Then $M$ is stable w.r.t. $S$.

**Time complexity analysis.** In the implementation, the preference order of a node $i \in V_S$ can be obtained by sorting (i.e., argsort) the $i^{th}$ row of $S$. Similarly, the preference order of a node $j \in V_T$ can be obtained by performing argsort operation on the $j^{th}$ column of $S$. Thus, the time complexity of obtaining the preference order of each node in $V_S$ and $V_T$ is $O(|V_S||V_T|\log(|V_T|))$. After deriving the preference orders, the remaining steps in the stable matching algorithm are performed with a time complexity of $O(|V_S|^2)$.

As a result, the time complexity of Algorithm \[\] is $O(|V_S||V_T|\log(|V_T|))$. In contrast, the Hungarian algorithm \cite{Kuhn1955}, which is commonly used to compute the final node correspondence between two graphs \cite{Wang2021, Gao2021, Yu2020, Saad-Eldin2021}, has a time complexity of $O(|V_T|^3)$.

It is worth noting that, in our stable matching algorithm, the column-wise and row-wise sorting operations can be parallelized when computing the preference orders of nodes. Therefore, in a parallelized setting, the time complexity of the stable matching algorithm can be further reduced.
Algorithm 1: Stable Matching algorithm

Input: A Sinkhorn Ranking S
Output: M

1. Obtain preference order for each node in $V_S$
2. Obtain preference order for each node in $V_T$
3. while $\exists i \in V_S$ that is not-matched ($M_{i,k} = 0 \ \forall k \in V_T$) do
4. $j =$ the highest ranked node in $i$’s preference order
5. if $j$ is not-matched ($M_{k,j} = 0 \ \forall k \in V_S$) then
6. $M_{i,j} \leftarrow 1$
7. else
8. if $i$ is ranked higher than $j$’s in the preference order of $j$ then
9. $M_{i',j} \leftarrow 0$
10. $M_{i,j} \leftarrow 1$
11. remove $j$ from the preference order of $i$
12. Return $M$

4 CONTRASTIVE MATCHING

In this section, we formulate a contrastive objective for graph matching. The key idea is to minimize a contrastive loss between positive samples (i.e., nodes that should be matched) and negative samples (i.e., nodes that should not be matched) of graph matching. However, despite the success of contrastive learning in many other domains, graph matching raises new challenges:

1. Although it is natural to regard each true-matching pair as a positive sample and each non-matching pair as a negative sample, it is unclear how to capture the relationship between positive samples and negative samples into a contrastive loss.

2. It is important to come up with a strategy to identify hard negative samples, as contrastive learning with an explicit hard negative sampling strategy has often shown to produce better results [Wang and Liu 2021].

We address these challenges in this section.

4.1 HARDNESS ATTENTION

We first design a hard negative sampling strategy to select hard negative samples according to their relatedness and informativeness to true matches.

Let $\pi : V_S \rightarrow V_T$ denote an injective function that maps every node $i \in V_S$ to a node $\pi(i) \in V_T$ as per the ground truth, i.e. $\pi(i)$ is the true match of $i$. We ground our design of a hard negative sampling strategy on two observations: (1) Relatedness: Randomly selecting negative samples is hardly effective for graph matching. This is because, for a true match $(i, \pi(i))$, its corresponding $S_{i,\pi(i)}$ has little correlation with other $S_{i',\pi(i)}$ if $i' \neq i$ and $j' \neq \pi(i)$. However, if considering $S_{ij'}$ with $j' \neq \pi(i)$ or $S_{i'j}$ with $i' \neq i$, they are closely correlated with $S_{i,\pi(i)}$ due to the constraints $S_{1n} = 1_m$ and $S_{1m} \leq 1_n$ imposed on rectangular doubly-stochastic matrices in $S$. Thus, node pairs corresponding to $S_{ij'}$ with $j' \neq \pi(i)$ or $S_{i'j}$ with $i' \neq i$ may serve as negative samples of $S_{i,\pi(i)}$. (2) Informativeness: Negative samples are often not equally informative. Specifically, among node pairs corresponding to $S_{ij'}$ with $j' \neq \pi(i)$ or $S_{i'j}$ with $i' \neq i$, some may be more informative than the others with respect to $S_{i,\pi(i)}$.

For each node $i \in V_S$, we define a hardness attention matrix $Z^{(i)} \in \{0, 1\}^{m \times n}$ as

$$Z_{k,j}^{(i)} = \begin{cases} 1, & \text{if } k = i, j \neq \pi(i) \text{ and } S_{k,j} \geq S_{i,\pi(i)} - \beta, \\ 1, & \text{if } k \neq i, j = \pi(i) \text{ and } S_{k,j} \geq S_{i,\pi(i)} - \beta, \\ 0, & \text{otherwise} \end{cases}$$

Here, $\beta \in [0, 1]$ is a parameter that regulates the degree of hardness of negative samples to be selected. $Z_{k,j}^{(i)}$ determines whether the $k$-th node in $G_S$ and the $j$-th node in $G_T$ should be selected as a hard negative sample with respect to the positive pair $(i, \pi(i))$ for training.

4.2 CONTRASTIVE MATCHING LOSS

Given a Sinkhorn ranking $S$ and a set of hardness attention matrices $Z = \{Z^{(i)}\}_{i \in V_S}$, we formulate a contrastive matching loss as:

$$L = \frac{1}{m} \sum_{i=1}^{m} -\frac{1}{2} \ln \left( \frac{f^+(i)}{1 + f^-(i)} \right)$$

where

$$f^+(i) = S_{i,\pi(i)}^2;$$

$$f^-(i) = \sum_{k=1}^{m} \sum_{j=1}^{n} Z_{k,j}^{(i)} S_{k,j}^2.$$

In these equations, $f^+(i)$ is the affinity of a positive sample associated with $i \in V_S$ and $f^-(i)$ is the sum of the affinities of corresponding hard negative samples, determined by the hardness attentions in $Z^{(i)}$. For clarity, we use $L_i$ to denote the contrastive loss for each $i \in V_S$:

$$L_i = -\frac{1}{2} \ln \left( \frac{f^+(i)}{1 + f^-(i)} \right)$$

This contrastive loss exhibits the following properties:

- For two nodes $\{i, i'\} \subseteq V_S$, if their sums of the affinities of the corresponding negative samples are the same, i.e. $f^-(i) = f^-(i')$, their losses are negatively correlated to the affinities of their positive samples.
We first analyze the gradients of our contrastive loss function. Accordingly, for a negative sample \(f^-(i) \neq f^+(i')\), if \(f^+(i) > f^+(i')\), then \(L_i > L_{i'}\).

- For two nodes \(\{i, i'\} \subseteq V_S\), if the affinities of their positive samples are the same, i.e. \(f^+(i) = f^+(i')\), their losses are positively correlated to the sums of the affinities of their negative samples.

**Property 1** (Informativeness of positive samples). Let \(\{i, i'\} \subseteq V_S\) with \(f^-(i) = f^-(i')\). If \(f^+(i') > f^+(i)\), then \(L_i > L_{i'}\).

**Property 2** (Informativeness of negative samples). Let \(\{i, i'\} \subseteq V_S\) with \(f^+(i) = f^+(i')\). If \(f^-(i') > f^-(i)\), then \(L_i > L_{i'}\).

A loss function \(L(\cdot)\) is said to be **positive-informative** or **negative-informative** if it satisfies Property 1 and Property 2, respectively. The following theorem can be easily proven. The proof is included in the supplementary material.

**Theorem 2.** The contrastive loss of Eq. 9 is both positive-informative and negative-informative.

It should be noted that given a positive sample \((i, \pi(i))\) s.t. \(i \in V_S\), it is possible that the sum of affinities of the related negative samples \(f^-(i)\) reaches 0 when \(Z_{i,j}^{(i)} = 0 \forall k \in V_S\) and \(\forall j \in V_T\). In such a situation, the standard InfoNCE contrastive loss [Wang and Liu 2021] is not positive informative, which is undesirable for graph matching.

### 4.3 Gradient Analysis

We first analyze the gradients of our contrastive loss function defined in Eq. 9.

Given any \(i \in V_S\), the gradient of the contrastive matching loss with respect to its positive sample is,

\[
\frac{\partial L_i}{\partial S_{i,\pi(i)}} = -\frac{1}{S_{i,\pi(i)}}. \tag{10}
\]

Accordingly, for a negative sample of \(i \in V_S\), the gradient of the contrastive matching loss with respect to \(S_{i,q}\), where \(q \neq \pi(i)\), is

\[
\frac{\partial L_i}{\partial S_{i,q}} = \frac{Z_{i,q}^{(i)} S_{i,q}}{1 + \sum_{k=1}^{m} \sum_{j=1}^{n} Z_{k,j}^{(i)} S_{k,j}^{2}}. \tag{11}
\]

Let \(\{i, i'\} \subseteq V_S\) and \(\{q, q'\} \subseteq V_T\). From Eq. 10 and Eq. 11 we have the following results:

- If \(S_{i,\pi(i)} > S_{i',\pi(i')}\), then \(\frac{\partial L_i}{\partial S_{i,\pi(i)}} > \frac{\partial L_{i'}}{\partial S_{i',\pi(i')}}\).

- If \(Z_{i,q}^{(i)} S_{i,q} > Z_{i',q'}^{(i')} S_{i',q'}\), then \(\frac{\partial L_i}{\partial S_{i,q}} > \frac{\partial L_{i'}}{\partial S_{i',q'}}\).

Eq. 10 and Eq. 12 show that a higher penalty is given to a positive sample with a lower affinity. Further, Eq. 11 and Eq. 13 show that the gradient w.r.t. a negative sample is hardness aware as hard negatives with higher affinities are penalized more than negative samples with lower affinities.

**Remark 1.** From the gradient analysis perspective, the standard contrastive loss function defined in [Wang and Liu 2021] has some undesirable behaviors in the graph matching setting. Specifically, because the magnitude of its gradients with respect to a positive sample is equal to the sum of the gradients of all the negative samples that are considered, a gradient will not be propagated if the total gradients from negative samples reach 0, even though the affinity of a positive sample may be further increased.

### 5 Further Discussion

Below, we discuss the optimality of matching achieved by our proposed method StableGM.

**Definition 3 (Optimal matching).** Let \(S\) be a Sinkhorn ranking. A matching \(M\) is \(G_S\)-optimal w.r.t. \(S\) if \(M_{i,j} = 1\) implies \(S_{i,j} > S_{i,k}\) for \(\forall k \in V_T \setminus \{j\}\). Similarly, a matching \(M\) is \(G_T\)-optimal w.r.t. \(S\) if \(M_{i,j} = 1\) implies \(S_{i,j} > S_{k,j}\) for \(\forall k \in V_S \setminus \{i\}\).

**Definition 4 (Dual-optimal matching).** A matching \(M\) is dual-optimal w.r.t. \(S\) if and only if \(M\) is both \(G_S\)-optimal and \(G_T\)-optimal w.r.t. \(S\).

**Definition 5 (R1-symmetry).** A Sinkhorn ranking \(S\) is R1-symmetric if following conditions are both satisfied.

- for any \(i \in V_S\) there exists some \(j \in V_T\) such that \(S_{i,j} > S_{i,k}\) holds for all \(k \in V_T \setminus \{j\}\);
- if \(S_{i,j} > S_{i,k}\) holds for all \(k \in V_T \setminus \{j\}\), then \(S_{i,j} > S_{k,j}\) also holds for all \(k \in V_S \setminus \{i\}\).

Based on the definition of R1-symmetry of \(S\), we can prove the following theorem and proposition. The proof details are included in the supplementary material.

**Theorem 3.** Let \(M\) be a matching produced by StableGM. Then \(M\) is dual-optimal if \(S\) is R1-symmetric.

**Proposition 1.** Let \(M\) be a matching produced by StableGM. Then \(M\) is dual-optimal when the contrastive loss function \(L(\cdot)\) is minimized.

It should be noted that, in its original setting, the MOSM algorithm for stable marriage assignment [McVitie and Wilson 1970] can only guarantee the optimality of solutions for one side, which is well known as the male optimality of stable marriage solutions [Gusfield and Irving 1989]. This is an undesired property for the graph matching problem since the optimality of a graph matching should be dual in
relation to both source and target graphs. Nonetheless, as shown in Proposition 1, in a contrastive learning setting with a properly designed loss function, a model can be learned to achieve dual-optimality of graph matching using the Stable Matching algorithm.

6 EXPERIMENTS

6.1 IMAGE KEYPOINTS MATCHING

Datasets We conduct experiments on three image-keypoint matching benchmarks widely used for learning-based graph matching: 1) Pascal VOC Keypoint with Berkeley annotations [Everingham et al., 2010], which consists of images belonging to 20 classes with keypoints annotated. 2) SPair-71k [Min et al., 2019], which contains 70958 image pairs from Pascal VOC 2012 and Pascal 3D+ belonging to 18 classes. 3) Willow ObjectClass [Cho et al., 2013] contains images that belong to 5 classes and images belonging to a class contain the same number of keypoints annotated.

Baselines We considered the following baselines: GMN [Zanfir and Sminchisescu, 2018], PCA-GM [Wang et al., 2019], IPCA-GM [Wang et al., 2020a], CIE [Yu et al., 2020], QC-DGM2 [Gao et al., 2021], BBGM [Rolínek et al., 2020], NGM-v2 [Wang et al., 2020b], NHGM-v2 [Wang et al., 2020d], and GCAN [Jiang et al., 2022].

Experimental setup For Pascal VOC Keypoint dataset, Willow ObjectClass dataset, and SPair-71k dataset, we followed the experimental setup in [Wang et al., 2021]. It is worth noting that GCAN [Jiang et al., 2022] followed the same experimental protocols as [Wang et al., 2021] for Pascal VOC Keypoint dataset and Willow ObjectClass dataset. However, for the SPair-71k dataset, GCAN [Jiang et al., 2022] followed the experimental setup of [Rolínek et al., 2020], which filters out keypoints that are outside of the bounding box of an image [Wang et al., 2021]. To consider keypoints that are outside of the bounding box of an image, we thus followed the experimental setup in [Wang et al., 2021].

In order to extract visual features of annotated keypoints, we followed the same feature extraction method as suggested in [Wang et al., 2021], where the feature vectors corresponding to each keypoint were obtained by concatenating the outputs of $\text{relu}_{42}$ and $\text{relu}_{51}$ operations of a pre-trained VGG16 model [Simonyan and Zisserman, 2014]. Then, for the input image pair, a graph corresponding to one annotated image was created by performing Delaunay triangulation [Delaunay et al., 1934] on keypoints and the graph corresponding to the other annotated image is fully connected on keypoints, following the same protocols as in [Wang et al., 2021].

In evaluations, matching accuracy is computed as the percentage of correct matchings among all true matchings [Wang et al., 2021].

StableGM models We evaluate two variants of StableGM: StableGM-v1 and StableGM-v2. Both variants are trained using the StableGM framework described in Section 3. They only differ in using techniques to learn node embeddings and to derive cross-graph node-to-node affinities.

In StableGM-v1, the neural architecture proposed in BBGM [Rolínek et al., 2020] is used as the neural network $\text{NN}_\theta$ to learn node embeddings, where node embeddings of the key points are obtained using a two-layer SplineCNN [Fey and Lenssen, 2019]. Moreover, in StableGM-v1, we use the affinity layer proposed in BBGM [Rolínek et al., 2020] to derive the cross-graph node-to-node affinity matrix $C$, where node affinities are obtained using a weighted inner product.

In StableGM-v2, we adapt the Graph Context Attention (GCA) mechanism proposed in GCAN [Jiang et al., 2022] to learn node embeddings and to obtain the cross-graph node-to-node affinity matrix. More specifically, after extracting the keypoint features of images, the self attention layer proposed in GCAN is used to learn node embeddings that encode graph structural and positional information. Then, the cross attention layer proposed in GCAN [Jiang et al., 2022] is used as the affinity layer to obtain cross-graph node-to-node affinities.

6.2 PPI NETWORK MATCHING

Datasets We further considered a dataset of protein-protein interaction (PPI) network of yeasts with its noisy versions. This PPI network consists of 1004 proteins and 4920 high-confidence interactions among proteins [Liu et al., 2021]. We match the PPI network with its three noisy versions, which contain 5%, 15%, and 25% low-confidence interactions, respectively, in addition to high-confidence interactions [Saraph and Milenkovic, 2014]. This dataset has been used by previous studies [Xu et al., 2019b; Liu et al., 2021] in evaluating graph matching approaches.

Baselines We considered the following baselines: PISwap [Chindelevitch et al., 2013], GHOST [Patro and Kingsford, 2012], MI-GRAAL [Kuchaiev and Pržulj, 2011], MAGNA++ [Vijayan et al., 2015], HubAlign [Hashemifar and Xu, 2014], NETAL [Neyshabur et al., 2013], CPD+Emb [Grover and Leskovec, 2016], Myronenko and Song, 2010], GWL+Emb [Xu et al., 2019b], GWL [Xu et al., 2019a], S-GWL [Xu et al., 2019a], and SIGMA [Liu et al., 2021].

Experimental setup To conduct a fair comparison, we follow the setup proposed in SIGMA [Liu et al., 2021] and S-GWL [Xu et al., 2019a]. Following the setup of SIGMA, the input feature of each node is assigned based on its node
Table 1: Matching accuracy (%) results on Pascal VOC Keypoint. Best results are in bold. Second best results are underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMN</td>
<td>61.8</td>
<td>71.2</td>
<td>77.6</td>
<td>78.7</td>
<td>83.5</td>
<td>83.3</td>
<td>79.4</td>
<td>75.4</td>
<td>86.3</td>
<td>79.9</td>
<td>75.3</td>
<td>78.5</td>
<td>81.4</td>
<td>93.2</td>
<td>79.7</td>
<td>86.1</td>
<td>79.9</td>
<td>92.4</td>
<td>94.8</td>
<td>88.8</td>
</tr>
<tr>
<td>PCA-GM</td>
<td>64.7</td>
<td>71.3</td>
<td>78.1</td>
<td>75.1</td>
<td>81.3</td>
<td>82.3</td>
<td>78.1</td>
<td>74.1</td>
<td>85.2</td>
<td>80.4</td>
<td>79.3</td>
<td>82.1</td>
<td>84.6</td>
<td>95.7</td>
<td>84.7</td>
<td>88.4</td>
<td>84.9</td>
<td>92.8</td>
<td>95.3</td>
<td>90.0</td>
</tr>
<tr>
<td>IPCA-GM</td>
<td>69.2</td>
<td>69.8</td>
<td>80.7</td>
<td>82.3</td>
<td>82.4</td>
<td>81.7</td>
<td>77.6</td>
<td>73.7</td>
<td>86.5</td>
<td>80.5</td>
<td>79.7</td>
<td>83.8</td>
<td>86.2</td>
<td>97.1</td>
<td>90.6</td>
<td>93.9</td>
<td>97.2</td>
<td>94.8</td>
<td>97.7</td>
<td>92.9</td>
</tr>
<tr>
<td>CIE-H</td>
<td>71.5</td>
<td>57.1</td>
<td>81.9</td>
<td>56.7</td>
<td>64.9</td>
<td>68.8</td>
<td>75.5</td>
<td>69.3</td>
<td>85.3</td>
<td>69.3</td>
<td>72.6</td>
<td>72.4</td>
<td>79.8</td>
<td>94.7</td>
<td>97.5</td>
<td>89.3</td>
<td>91.8</td>
<td>95.9</td>
<td>98.0</td>
<td>93.2</td>
</tr>
<tr>
<td>BBGM</td>
<td>79.5</td>
<td>66.1</td>
<td>67.9</td>
<td>67.4</td>
<td>67.4</td>
<td>67.1</td>
<td>67.1</td>
<td>67.3</td>
<td>67.4</td>
<td>67.1</td>
<td>67.3</td>
<td>67.1</td>
<td>67.3</td>
<td>67.1</td>
<td>67.3</td>
<td>67.1</td>
<td>67.1</td>
<td>67.1</td>
<td>67.1</td>
<td>67.1</td>
</tr>
<tr>
<td>NGM-v2</td>
<td>59.9</td>
<td>71.7</td>
<td>77.2</td>
<td>79.0</td>
<td>87.7</td>
<td>81.8</td>
<td>80.0</td>
<td>71.6</td>
<td>79.2</td>
<td>79.2</td>
<td>80.7</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
<td>81.8</td>
</tr>
<tr>
<td>GCAN</td>
<td>63.4</td>
<td>70.1</td>
<td>80.1</td>
<td>81.1</td>
<td>80.4</td>
<td>81.8</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
<td>80.4</td>
</tr>
</tbody>
</table>

As shown in Table 1, StableGM-v2 outperforms all other baselines in seven out of twenty classes of the Pascal VOC Keypoint dataset. Specifically, while our approach outperforms baselines for some classes (e.g., aero, sheep, bike), it performs under par for some other classes. When observing the results on the SPair-71k dataset in Table 2 both StableGM-v1 and StableGM-v2 outperform all other baselines when it comes to mean accuracy over all the classes. It should be noted that, compared with the Pascal VOC Keypoint dataset, the SPair-71k dataset possesses several advantages, such as higher image quality, more reliable keypoint annotations, and removal of sofa and table class which are ambiguous and poorly annotated [Rolínek et al., 2020]. Indeed, it can be seen from Table 1 that the mean accuracy of StableGM-v2 is less than GCAN primarily because StableGM-v2 performs poorly for sofa and table classes.

For the Willow ObjectClass dataset, Table 3 shows that the mean accuracy results of StableGM models and GCAN are both high, close to 100. Their performance is generally comparable. The mean accuracy of GCAN is slightly higher than our models. However, StableGM-v2 outperforms GCAN in 2 classes while performing equally well in 1 class; StableGM-v1 outperforms GCAN in 2 classes while performing equally well in 2 classes.

6.3 RESULTS AND DISCUSSION

StableGM model To evaluate our approach on the PPI network, the StableGM model was initialized as follows. A 5-layer Graph Isomorphism Network (GIN) was used as NN0 to learn node embeddings. It should be noted that the same graph neural network architecture was used in SIGMA [Liu et al., 2021] to learn node embeddings. In the affinity layer of the StableGM model, we employ the same technique used in SIGMA for a fair comparison, which calculates the affinity between two nodes as the cosine similarity between the embeddings of the two nodes.

degree. We use node correctness as the evaluation metric, calculated as the percentage of nodes that have the same matching as the ground truth [Liu et al., 2021; Xu et al., 2019a].
Table 4: Node correctness (%) results on the PPI dataset. The best results are in bold. The second best results are underlined.

<table>
<thead>
<tr>
<th>Method</th>
<th>Yeast 5%</th>
<th>Yeast 15%</th>
<th>Yeast 25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PISwap</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>GHOST</td>
<td>11.1</td>
<td>0.4</td>
<td>0.3</td>
</tr>
<tr>
<td>MI-GRAAL</td>
<td>18.0</td>
<td>6.9</td>
<td>5.2</td>
</tr>
<tr>
<td>MAGNA++</td>
<td>48.1</td>
<td>25.0</td>
<td>13.6</td>
</tr>
<tr>
<td>HubAlign</td>
<td>50.0</td>
<td>35.2</td>
<td>12.9</td>
</tr>
<tr>
<td>NETAL</td>
<td>6.9</td>
<td>0.9</td>
<td>1.0</td>
</tr>
<tr>
<td>CFD+Emb</td>
<td>3.6</td>
<td>2.1</td>
<td>2.0</td>
</tr>
<tr>
<td>GWL+Emb</td>
<td>83.7</td>
<td>66.6</td>
<td>58.0</td>
</tr>
<tr>
<td>GWL</td>
<td>82.4</td>
<td>65.35</td>
<td>58.8</td>
</tr>
<tr>
<td>S-GWL</td>
<td>81.1</td>
<td>61.85</td>
<td>56.27</td>
</tr>
<tr>
<td>SIGMA</td>
<td>84.7±0.4</td>
<td>57.4±1.1</td>
<td>41.4±1.7</td>
</tr>
<tr>
<td>StableGM</td>
<td>66.4±0.9</td>
<td>67.5±1.1</td>
<td>57±0.6</td>
</tr>
</tbody>
</table>

We have further carried out experiments to demonstrate the effect of selecting the hardness attention parameter $\beta$ and the results are included in the supplementary material.

7 RELATED WORK

Deep Graph Matching In the literature of deep graph matching, the problem of graph matching has been primarily formulated in several ways: 1) Quadratic Assignment problem [Wang et al., 2019; Yu et al., 2020; Rolínek et al., 2020; Wang et al., 2021], 2) Stochastic Optimization Problem [Liu et al., 2021], 3) Optimal Transport Problem [Yu et al., 2020], and 4) Integer Linear Programming Problem [Jiang et al., 2022]. The study in [Zanfir and Sminchisescu, 2018] is considered the first attempt to use deep learning to solve graph matching. Graph Neural Networks (GNNs) were first employed for graph matching in the seminal work [Wang et al., 2019]. The use of GNNs enables the formulation of graph matching as a linear assignment problem which is then solved using the Hungarian algorithm. BBGM [Rolínek et al., 2020] considered graph matching as a Quadratic Assignment Problem and adapted the technique of black box differentiation of combinatorial solvers [Vlastelica et al., 2019] to obtain the final node correspondence. The representation learning technique proposed in [Rolínek et al., 2020] for graph matching in keypoint annotated image datasets have been used in [Wang et al., 2021] to achieve the state-of-the-art results while considering graph matching as a constrained vertex classification task.

Contrastive Learning A contrastive loss function was introduced in [Chopra et al., 2005], in which the objective is to minimize the Euclidean distance between the feature vectors of positive samples as much as possible while increasing that of the negative samples more than a pre-defined distance. Subsequently, the concept of contrastive learning has been extended to different applications and various contrastive loss functions such as Triplet loss [Schroff et al., 2015], N-pair loss [Sohn, 2016] and InfoNCE loss [Oord et al., 2018] have been introduced. Recently, contrastive learning has attracted significant attention due to its success in self-supervised learning [Wu et al., 2018; He et al., 2020; Wang and Liu, 2021]. It has been shown that the performance of contrastive losses is enhanced when selecting explicit hard negative sampling [Xu et al., 2022]. In [Khosla et al., 2020], the standard contrastive loss function has been adapted to the supervised classification setting by incorporating the information related to class labels. The main difference of this setting, when compared with self-supervised contrastive learning, is the availability of many positive samples per anchor (in addition to the many negative samples), because the data instances from the same class are considered as positives.

8 CONCLUSION

In this work, we introduced a new contrastive learning framework for graph matching, named StableGM. Within our contrastive learning framework, we proposed a novel hard negative sampling strategy and a new contrastive loss function that suits the graph matching setting. We provide a theoretical analysis of the proposed contrastive matching loss and showed that the proposed loss function possesses properties that can overcome the limitations that can occur when adapting standard contrastive loss function to our setting. Additionally, we described how our StableGM framework provides theoretical guarantees for optimal matching. We conducted experiments on benchmark datasets that are widely used for deep graph matching, and our empirical evaluation demonstrates the effectiveness of our approach. This work opens the direction to consider supervised graph matching as a contrastive learning problem and thereby adapt the concepts discussed in contrastive learning literature to the graph matching domain.

Our code is available at GitHub: https://github.com/Gathika94/StableGM.git

Acknowledgements

This work was partly supported by the Australian Research Council under Discovery Project DP210102273. We also would like to thank anonymous reviewers for their comments which helped improve the quality of the paper.
References


