Contrastive Learning for Supervised Graph Matching

Gathika Ratnayaka[1] **Qing Wang**[1] **Yang Li**[2]

[1]School of Computing, Australian National University, Canberra, Australia
[2]School 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, Loiola 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

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

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 $\mathcal{G} = (V, E, \mathbf{A}, \mathbf{X})$, where $V = \{1, \dots, |V|\}$ is the set of vertices, E is the set of edges, $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{|V|\times d}$ is a node feature matrix with feature vectors of dimension d. Given two graphs, a source graph $\mathcal{G}_S = (V_S, E_S, \mathbf{A}_S, \mathbf{X}_S)$ and a target graph $\mathcal{G}_T = (V_T, E_T, \mathbf{A}_T, \mathbf{X}_T)$, 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 $\mathbf{M} \in \{0, 1\}^{m\times n}$ between $\mathcal{G}_S$ and $\mathcal{G}_T$ which satisfies the one-to-one mapping constraint $\sum_{j=1}^{n} \mathbf{M}_{i,j} = 1 \; \forall i \in V_S$ and $\sum_{i=1}^{m} \mathbf{M}_{i,j} \leq 1 \; \forall j \in V_T$ and is optimal w.r.t. an objective function $f(\cdot)$:

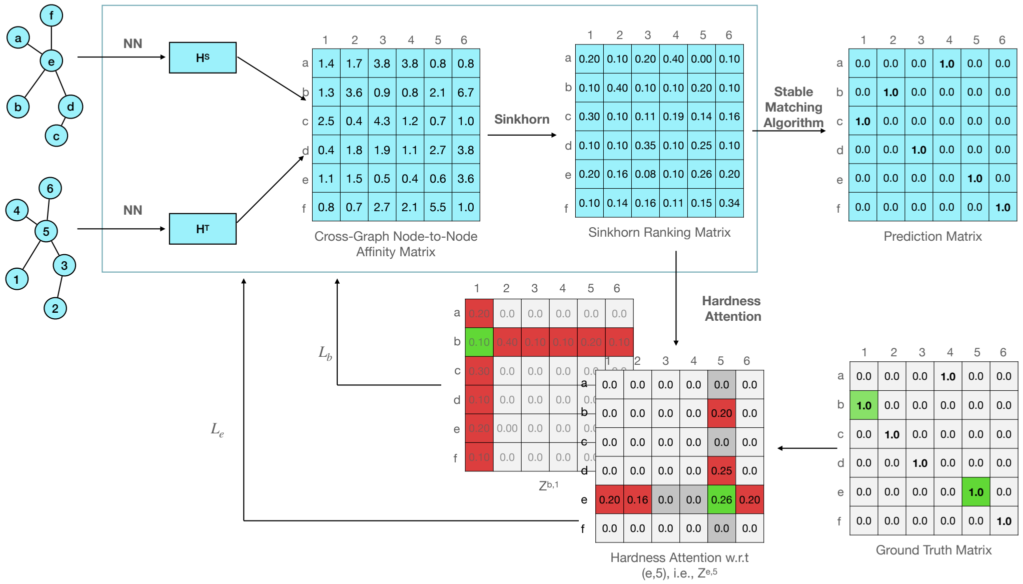
$$\arg\min_{\mathbf{M}} f(\mathbf{M}; \mathcal{G}_S, \mathcal{G}_T). \tag{1}$$

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

$$\arg\min_{\mathbf{M}} L(\mathbf{M}^*, \mathbf{M}; \mathcal{G}_S, \mathcal{G}_T). \tag{2}$$

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

Problem formulation Let $\mathbb{G}$ denote a set of graphs on a finite number of nodes and $\mathbb{M}$ be the set of $m \times n$ matching matrices, i.e., $\mathbb{M} = \{\mathbf{M} \in \{0, 1\}^{m\times n} | \sum_{j=1}^{n} \mathbf{M}_{i,j} = 1 \forall i \in V_S$ and $\sum_{i=1}^{m} \mathbf{M}_{i,j} \leq 1 \; \forall j \in V_T$. In this work, our goal is to develop a deep graph matching model $f_\Theta : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{M}$ by end-to-end training through optimizing a contrastive matching objective.



	1	2	3	4	5	6
a	1.4	1.7	3.8	3.8	0.8	0.8
b	1.3	3.6	0.9	0.8	2.1	6.7
c	2.5	0.4	4.3	1.2	0.7	1.0
d	0.4	1.8	1.9	1.1	2.7	3.8
e	1.1	1.5	0.5	0.4	0.6	3.6
f	0.8	0.7	2.7	2.1	5.5	1.0

Cross-Graph Node-to-Node Affinity Matrix

	1	2	3	4	5	6
a	0.20	0.10	0.20	0.40	0.00	0.10
b	0.10	0.40	0.10	0.10	0.20	0.10
c	0.30	0.10	0.11	0.19	0.14	0.16
d	0.10	0.10	0.35	0.10	0.25	0.10
e	0.20	0.16	0.08	0.10	0.26	0.20
f	0.10	0.14	0.16	0.11	0.15	0.34

Sinkhorn Ranking Matrix

	1	2	3	4	5	6
a	0.0	0.0	0.0	**1.0**	0.0	0.0
b	0.0	**1.0**	0.0	0.0	0.0	0.0
c	**1.0**	0.0	0.0	0.0	0.0	0.0
d	0.0	0.0	**1.0**	0.0	0.0	0.0
e	0.0	0.0	0.0	0.0	**1.0**	0.0
f	0.0	0.0	0.0	0.0	0.0	**1.0**

Prediction Matrix

	1	2	3	4	5	6
a	0.20	0.0	0.0	0.0	0.0	0.0
b	0.10	0.40	0.10	0.10	0.20	0.10
c	0.30	0.0	0.0	0.0	0.0	0.0
d	0.10	0.0	0.0	0.0	0.0	
e	0.20	0.00	0.0	0.0	0.0	
f	0.10	0.0	0.0	0.0	0.0	

$Z^{b,1}$

	1	2	3	4	5	6
a	0.0	0.0	0.0	0.0	0.0	0.0
b	0.0	0.0	0.0	0.0	0.20	0.0
c	0.0	0.0	0.0	0.0	0.0	0.0
d	0.0	0.0	0.0	0.0	0.25	0.0
e	0.20	0.16	0.0	0.0	0.26	0.20
f	0.0	0.0	0.0	0.0	0.0	0.0

Hardness Attention w.r.t (e,5), i.e., $Z^{e,5}$

	1	2	3	4	5	6
a	0.0	0.0	0.0	**1.0**	0.0	0.0
b	**1.0**	0.0	0.0	0.0	0.0	0.0
c	0.0	**1.0**	0.0	0.0	0.0	0.0
d	0.0	0.0	**1.0**	0.0	0.0	0.0
e	0.0	0.0	0.0	0.0	**1.0**	0.0
f	0.0	0.0	0.0	0.0	0.0	**1.0**

Ground Truth Matrix

Figure 1: Overview of the StableGM framework. Given two graphs, their node embeddings are obtained using a neural network (NN) and a cross-graph node-to-node affinity matrix is computed. Then, Sinkhorn normalization is performed to obtain a Sinkhorn ranking matrix. In the training phase, the Sinkhorn ranking matrix is compared with the ground truth matrix and hard negative sampling is performed via "Hardness Attention", e.g., a positive sample is highlighted in green while the corresponding negative samples are highlighted in red. In the test phase, the Stable Matching algorithm takes the Sinkhorn ranking matrix as input and yields the node-to-node correspondence between the two graphs.

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. $\mathbb{S} = \{\mathbf{S} \in [0,1]^{m \times n} | \mathbf{S}\mathbf{1}_n = \mathbf{1}_m, \mathbf{S}^\intercal \mathbf{1}_m \leq \mathbf{1}_n\}$.

StableGM consists of three main components.

(1) *Graph affinity encoding*: a permutation-equivariant, differentiable function $\psi_{encode} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{C}$ 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 $\psi_{rank} : \mathbb{C} \rightarrow \mathbb{S}$ 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 $\psi_{match} : \mathbb{S} \rightarrow \mathbb{M}$ 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 θ, 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 NN_θ to obtain the node embedding matrices of $\mathcal{G}_S$ and $\mathcal{G}_T$, respectively:

$$\text{NN}_\theta(\mathcal{G}_S) = \mathbf{H}^S \in \mathbb{R}^{m \times d} \,;\, \text{NN}_\theta(\mathcal{G}_T) = \mathbf{H}^T \in \mathbb{R}^{n \times d}. \quad (3)$$

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

$$\mathbf{C} = f_\gamma(\mathbf{H}^S, \mathbf{H}^T) \quad (4)$$

Here, f_γ denotes an affinity layer parameterized by γ that encodes the cross-graph node-to-node affinities between $\mathcal{G}_S$ and $\mathcal{G}_T$. Different techniques such as the inner product [Fey et al., 2020], and the weighted inner product [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 [Sinkhorn, 1964] to compute the ranking of nodes in terms of their preference orders for matching.

Following [Wang et al., 2020b], a nonnegative matrix $\mathbf{S}^0$ is first obtained via $\mathbf{S}^0_{i,j} = \exp(\mathbf{C}_{ij}/\alpha)$, where $\alpha > 0$ is a parameter. In the case where the target graph is larger, i.e., $m < n$, the matrix $\mathbf{S}^0$ is padded into a square matrix with small positive values (e.g. 1^{-10}). Then, the rows and columns of $\mathbf{S}^0$ are iteratively normalized into a rectangular doubly-stochastic matrix using an extended Sinkhorn algorithm [Cour et al., 2007, Fey et al., 2020]. Specifically, let $\eta(\cdot)$ be the Sinkhorn operator, $\oslash$ be the elementwise Hadamard division, and $\mathbf{1_n}$ be a column vector of ones. Starting with $\eta^0(\hat{\mathbf{S}}) = \mathbf{S}^0$, we have

$$\eta^i(\hat{\mathbf{S}}) = f_{row}(f_{col}(\eta^{i-1}(\hat{\mathbf{S}}))), \tag{5}$$

where

$$f_{row}(\hat{\mathbf{S}}) = \hat{\mathbf{S}} \oslash (\hat{\mathbf{S}}\mathbf{1_n}\mathbf{1_n}^\intercal) \tag{6}$$

$$f_{col}(\hat{\mathbf{S}}) = \hat{\mathbf{S}} \oslash (\mathbf{1_n}\mathbf{1_n}^\intercal\hat{\mathbf{S}}). \tag{7}$$

Padded elements are discarded from the final output to obtain a rectangular doubly stochastic matrix $\mathbf{S} \in \mathbb{S}$. It should be noted that $\mathbf{S}_{i,j}$ 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 $\mathbf{S}_{i,j} > \mathbf{S}_{i,k}$, j is ranked higher than k in i's preference order. If $\mathbf{S}_{i,j} = \mathbf{S}_{i,k}$, 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 $\mathbf{S}_{i,j} > \mathbf{S}_{k,j}$, i is ranked higher than k in j's preference order. If $\mathbf{S}_{i,j} = \mathbf{S}_{k,j}$, the tie is broken arbitrary. As the preference orders of the nodes are derived based on $\mathbf{S}$, we term $\mathbf{S}$ 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 [Gale and Shapley, 1962, McVitie and Wilson, 1970] to efficiently generate graph matching. We begin with defining the notion of blocking pair.

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

- $\mathbf{M}_{i,j} = 0$*;*
- $\exists k \in V_S \backslash \{i\}$ $(\mathbf{M}_{k,j} = 1 \wedge \mathbf{S}_{i,j} > \mathbf{S}_{k,j})$*;*
- $\exists k' \in V_T \backslash \{j\}$ $(\mathbf{M}_{i,k'} = 1 \wedge \mathbf{S}_{i,j} > \mathbf{S}_{i,k'}) \vee \forall k' \in V_T \backslash \{j\}$ $\mathbf{M}_{i,k'} = 0$.

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. $\mathbf{M}$, $\mathbf{M}$ is considered as being unstable.

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

Algorithm 1 describes our stable matching algorithm, which extends the Male Optimal Stable Marriage (MOSM) algorithm proposed in [McVitie and Wilson, 1970] to the graph matching setting. Taking a Sinkhorn ranking $\mathbf{S}$ 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 $\mathbf{M} \in \{0, 1\}^{m \times n}$ that contains one-to-one node correspondences from V_S to V_T. Here, $\mathbf{M}_{i,j} = 1$ 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* $\mathbf{S}$*, Algorithm 1 can always return a matching* $\mathbf{M}$ *that satisfies the one-to-one mapping constraint:* $\sum_{j=1}^{n} \mathbf{M}_{i,j} = 1 \ \forall i \in V_S$ *and* $\sum_{i=1}^{m} \mathbf{M}_{i,j} \leq 1 \ \forall j \in V_T$.

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

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 $\mathbf{S}$. Similarly, the preference order of a node $j \in V_T$ can be obtained by performing *argsort* operation on the j^{th} column of $\mathbf{S}$. 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 1 is $O(|V_S||V_T|log(|V_T|))$. In contrast, the Hungarian algorithm [Kuhn, 1955], which is commonly used to compute the final node correspondence between two graphs [Wang et al., 2021, Gao et al., 2021, Yu et al., 2020, Saad-Eldin et al., 2021], 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 $\mathbf{S}$
Output: $\mathbf{M}$
Obtain preference order for each node in $V_S$
Obtain preference order for each node in $V_T$
while $\exists i \in V_S$ that is not-matched ($\mathbf{M}_{i,k} = 0$ $\forall k \in V_T$) do
    $j =$ the highest ranked node in $i$'s preference order
    if $j$ is not-matched ($\mathbf{M}_{k,j} = 0 \; \forall k \in V_S$) then
        $\mathbf{M}_{i,j} \leftarrow 1$
    else
        $i' =$ node that is currently matched with $j$
        if $i$ is ranked higher than $i'$ in the preference order of $j$ then
            $\mathbf{M}_{i',j} \leftarrow 0$
            $\mathbf{M}_{i,j} \leftarrow 1$
    remove $j$ from the preference order of $i$
Return $\mathbf{M}$
```

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 $\mathbf{S}_{i,\pi(i)}$ has little correlation with other $\mathbf{S}_{i',j'}$ if $i' \neq i$ and $j' \neq \pi(i)$. However, if considering $\mathbf{S}_{ij'}$ with $j' \neq \pi(i)$ or $\mathbf{S}_{i'\pi(i)}$ with $i' \neq i$, they are closely correlated with $\mathbf{S}_{i,\pi(i)}$ due to the constraints $\mathbf{S1}_n = \mathbf{1}_m$ and $\mathbf{S}^\intercal \mathbf{1}_m \leq \mathbf{1}_n$ imposed on rectangular doubly-stochastic matrices in $\mathbb{S}$. Thus, node pairs corresponding to $\mathbf{S}_{ij'}$ with $j' \neq \pi(i)$ or $\mathbf{S}_{i'\pi(i)}$ with $i' \neq i$ may serve as negative samples of $\mathbf{S}_{i,\pi(i)}$. (2) *Informativeness:* Negative samples are often not equally informative. Specifically, among node pairs corresponding to $\mathbf{S}_{ij'}$ with $j' \neq \pi(i)$ or $\mathbf{S}_{i'\pi(i)}$ with $i' \neq i$, some may be more informative than the others with respect to $\mathbf{S}_{i,\pi(i)}$.

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

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

Here, $\beta \in [0, 1]$ is a parameter that regulates the degree of hardness of negative samples to be selected. $\mathbf{Z}^{(i)}_{k,j}$ determines whether the k-th node in $\mathcal{G}_S$ and the j-th node in $\mathcal{G}_T$ 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 $\mathbf{S}$ and a set of hardness attention matrices $\mathbf{Z} = \{\mathbf{Z}^{(i)}\}_{i \in V_S}$, we formulate a contrastive matching loss as:

$$L = \frac{1}{m} \sum_{i=1}^{m} -\frac{1}{2} \ln \left(\frac{f^+(i)}{1 + f^-(i)} \right) \tag{8}$$

where

$$\begin{aligned} f^+(i) =& \mathbf{S}^2_{i,\pi(i)}; \\ f^-(i) =& \sum_{k=1}^{m} \sum_{j=1}^{n} \mathbf{Z}^{(i)}_{k,j} \mathbf{S}^2_{k,j}. \end{aligned}$$

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 $\mathbf{Z}^{(i)}$. For clarity, we use L_i to denote the contrastive loss for each $i \in V_S$:

$$L_i = -\frac{1}{2} \ln \left(\frac{f^+(i)}{1 + f^-(i)} \right) \tag{9}$$

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.

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

- 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 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 $\mathbf{Z}^{(i)}_{k,j} = 0 \, \forall k \in V_S$ 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,

$$\frac{\partial L_i}{\partial \mathbf{S}_{i,\pi(i)}} = -\frac{1}{\mathbf{S}_{i,\pi(i)}}. \tag{10}$$

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

$$\frac{\partial L_i}{\partial \mathbf{S}_{i,q}} = \frac{\mathbf{Z}^{(i)}_{iq}\mathbf{S}_{i,q}}{1 + \sum_{k=1}^{m}\sum_{j=1}^{n} \mathbf{Z}^{(i)}_{k,j}\mathbf{S}^2_{k,j}}. \tag{11}$$

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

- If $\mathbf{S}_{i,\pi(i)} < \mathbf{S}_{i',\pi(i')}$, then

$$|\frac{\partial L_i}{\partial \mathbf{S}_{i,\pi(i)}}| > |\frac{\partial L_{i'}}{\partial \mathbf{S}_{i',\pi(i')}}| \tag{12}$$

- If $\mathbf{Z}^{(i)}_{iq}\mathbf{S}_{i,q} > \mathbf{Z}^{(i)}_{iq'}\mathbf{S}_{i,q'}$, then

$$|\frac{\partial L_i}{\partial \mathbf{S}_{i,q}}| > |\frac{\partial L_i}{\partial \mathbf{S}_{i,q'}}| \tag{13}$$

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* $\mathbf{S}$ *be a Sinkhorn ranking. A matching* $\mathbf{M}$ *is* $\mathcal{G}_S$-optimal *w.r.t.* $\mathbf{S}$ *if* $\mathbf{M}_{i,j} = 1$ *implies* $\mathbf{S}_{i,j} > \mathbf{S}_{i,k}$ *for* $\forall k \in V_T \backslash \{j\}$*. Similarly, a matching* $\mathbf{M}$ *is* $\mathcal{G}_T$-optimal *w.r.t.* $\mathbf{S}$ *if* $\mathbf{M}_{i,j} = 1$ *implies* $\mathbf{S}_{i,j} > \mathbf{S}_{k,j}$ *for* $\forall k \in V_S \backslash \{i\}$*.*

Definition 4 (Dual-optimal matching)**.** *A matching* $\mathbf{M}$ *is* dual-optimal *w.r.t.* $\mathbf{S}$ *if and only if* $\mathbf{M}$ *is both* $\mathcal{G}_S$*-optimal and* $\mathcal{G}_T$*-optimal w.r.t.* $\mathbf{S}$*.*

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

- *for any* $i \in V_S$ *there exists some* $j \in V_T$ *such that* $\mathbf{S}_{i,j} > \mathbf{S}_{i,k}$ *holds for all* $k \in V_T \backslash \{j\}$*;*
- *if* $\mathbf{S}_{i,j} > \mathbf{S}_{i,k}$ *holds for all* $k \in V_T \backslash \{j\}$*, then* $\mathbf{S}_{i,j} > \mathbf{S}_{k,j}$ *also holds for all* $k \in V_S \backslash \{i\}$*.*

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

Theorem 3. *Let* $\mathbf{M}$ *be a matching produced by StableGM. Then* $\mathbf{M}$ *is dual-optimal if* $\mathbf{S}$ *is R1-symmetric.*

Proposition 1. *Let* $\mathbf{M}$ *be a matching produced by StableGM. Then* $\mathbf{M}$ *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., 2020b], 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][1]. 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, Jiang et al., 2022] where the feature vectors corresponding to each keypoint were obtained by concatenating the outputs of $relu_{42}$ and $relu_{51}$ operations of a pre-trained VGG$_{16}$ 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 Section 4. 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 NN_θ to learn node embeddings, where node embeddings of the key points are obtained using a two-layer SplineCNN [Fey et al., 2018]. 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 $\mathbf{C}$, 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 Milenković, 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

[1] https://github.com/Thinklab-SJTU/ThinkMatch

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
GMN	41.6	59.6	60.3	48.0	79.2	70.2	67.4	64.9	39.2	61.3	66.9	59.8	61.1	59.8	37.2	78.2	68.0	49.9	84.2	91.4	62.4
PCA-GM	49.8	61.9	65.3	57.2	78.8	75.6	64.7	69.7	41.6	63.4	50.7	67.1	66.7	61.6	44.5	81.2	67.8	59.2	78.5	90.4	64.8
IPCA-GM	53.8	66.2	67.1	61.2	80.4	75.3	72.6	72.5	44.6	65.2	54.3	67.2	67.9	64.2	47.9	84.4	70.8	64.0	83.8	90.8	67.7
CIE-H	52.5	68.6	70.2	57.1	82.1	77	70.7	73.1	43.8	69.9	62.4	70.2	70.3	66.4	47.6	85.3	71.7	64	83.9	91.7	68.9
qc-DGM2	49.6	64.6	67.1	62.4	82.1	79.9	74.8	73.5	43.0	68.4	66.5	67.2	71.4	70.1	48.6	92.4	69.2	70.9	90.9	92.0	70.3
BBGM	61.9	71.1	<u>79.7</u>	79.0	87.4	94.0	89.5	80.2	56.8	79.1	64.6	78.9	76.2	75.1	65.2	98.2	77.3	77.0	94.9	**93.9**	79.0
NGM-v2	61.8	71.2	77.6	78.8	87.3	93.6	87.7	79.8	55.4	77.8	89.5	78.8	80.1	**79.2**	62.6	97.7	77.7	75.7	96.7	93.2	80.1
NHGM-v2	59.9	71.5	77.2	79.0	87.7	94.6	89.0	**81.8**	60.0	<u>81.3</u>	<u>87.0</u>	78.1	76.5	77.5	64.4	**98.7**	<u>77.8</u>	75.4	97.9	92.8	80.4
GCAN	63.4	71.2	**80.1**	**81.1**	**90.4**	**95.5**	<u>89.5</u>	80.4	**65.3**	80.8	**89.9**	<u>81.4</u>	<u>80.6</u>	78.1	**67.7**	98.2	77.5	**82.6**	<u>98.4</u>	93.4	**82.3**
StableGM-v1	<u>63.57</u>	**72.85**	79.66	<u>81.02</u>	<u>88.73</u>	<u>94.71</u>	88.81	78.41	59.96	79.14	84.33	80.48	78.43	<u>78.45</u>	63.90	97.90	77.77	<u>78.25</u>	98.00	93.24	80.88
StableGM-v2	**65.07**	<u>72.51</u>	79.36	79.28	88.22	94.42	**89.98**	<u>81.3</u>	<u>65.05</u>	**81.34**	82.38	**82.25**	**81.24**	77.53	<u>66.11</u>	<u>98.36</u>	**79.44**	74.33	**98.47**	<u>93.71</u>	<u>81.52</u>

Table 1: Matching accuracy (%) results on Pascal VOC Keypoint. Best results are in bold. Second best results are underlined.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	mbike	person	plant	sheep	train	tv	mean
GMN	59.9	51	74.3	46.7	63.3	75.5	69.5	64.6	57.5	73	58.7	59.1	63.2	51.2	86.9	57.9	70	92.4	65.3
PCA-GM	64.7	45.7	78.1	51.3	63.8	72.7	61.2	62.8	62.6	68.2	59.1	61.2	64.9	57.7	87.4	60.4	72.5	92.8	66
IPCA-GM	69	52.9	80.4	54.3	66.5	80	68.5	71.4	61.4	74.8	66.3	65.1	69.6	63.9	91.1	65.4	82.9	97.5	71.2
CIE-H	71.5	57.1	81.7	56.7	67.9	82.5	73.4	74.5	62.6	78	68.7	66.3	73.7	66	92.5	67.2	82.3	97.5	73.3
BBGM	72.50	64.55	87.80	75.81	69.27	93.95	88.59	79.92	**74.56**	83.15	78.78	**77.10**	76.50	76.34	98.20	85.54	96.78	99.31	82.15
NGM-v2	68.8	63.3	86.8	70.1	69.7	94.7	87.4	77.4	72.1	80.7	74.3	72.5	79.5	73.4	98.9	81.2	94.3	98.7	80.2
NHGM-v2	62	57.8	86.4	68.5	68.7	93.4	80.8	76.6	69.2	79.9	66.2	71.7	78.1	69.5	98.2	84.4	93.2	99.3	78
GCAN	69.56	61.88	**89.85**	75.21	**70.41**	<u>97.22</u>	87.55	<u>80.39</u>	70.37	83.77	75.48	72.26	**81.17**	75.52	**99.71**	86.07	**97.75**	**99.94**	81.90
StableGM-v1	<u>74.20</u>	<u>65.78</u>	88.05	<u>76.46</u>	69.60	**97.70**	<u>90.18</u>	**80.99**	71.53	**85.18**	<u>78.13</u>	<u>74.61</u>	79.58	**79.13**	<u>99.7</u>	**87.09**	<u>97.53</u>	99.47	<u>83.05</u>
StableGM-v2	**75.08**	**68.65**	<u>89.50</u>	**76.86**	<u>70.14</u>	97.16	**91.74**	79.15	<u>72.58</u>	<u>84.56</u>	**81.58**	74.06	<u>80.85</u>	<u>78.93</u>	99.04	<u>86.81</u>	96.10	<u>99.76</u>	**83.48**

Table 2: Matching accuracy (%) results on SPair-71K. Best results are in bold. Second best results are underlined.

Method	car	duck	face	mbike	w-bottle	mean
GMN	67.9	76.7	99.8	69.2	83.1	79.3
PCA-GM	87.6	83.6	**100.0**	77.6	88.4	87.4
IPCA-GM	90.4	88.6	**100.0**	83.0	88.3	90.1
qc-DGM2	**100.0**	**98.8**	98.0	92.8	99.0	97.7
BBGM	96.8	89.9	**100.0**	99.8	<u>99.4</u>	97.2
NGM-v2	97.4	93.4	**100.0**	98.6	98.3	97.5
NHGM-v2	97.4	93.9	**100.0**	98.6	98.9	97.8
GCAN	98.8	94.1	**100.0**	**100.0**	**100**	**98.6**
StableGM-v1	<u>98.85</u>	94.23	**100.0**	**100.0**	99.23	98.46
StableGM-v2	<u>98.85</u>	<u>94.62</u>	**100.0**	99.81	99.31	<u>98.52</u>

Table 3: Matching accuracy (%) results on Willow ObjectClass. Best results are in bold. Second best results are underlined.

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].

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 NN_θ 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.

6.3 RESULTS AND DISCUSSION

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.

Table 4 shows that our StableGM model achieves superior performance compared to other baselines when matching the PPI network with its 5% and 15% noisy versions. However, when the noise level increases to 25%, the model's performance becomes inferior to GWL+Emb [Xu et al., 2019b] and GWL [Xu et al., 2019a]. These empirical find-

Method	Yeast 5%	Yeast 15%	Yeast 25%
PISwap	0.1	0.1	0.0
GHOST	11.1	0.4	0.3
MI-GRAAL	18.0	6.9	5.2
MAGNA++	48.1	25.0	13.6
HubAlign	50.0	35.2	12.9
NETAL	6.9	0.9	1.0
CPD+Emb	3.6	2.1	2.0
GWL+Emb	83.7	66.6	58.0
GWL	82.4	65.34	**58.8**
S-GWL	81.1	61.85	56.27
SIGMA	84.7±0.4	57.4±1.1	41.4 ±1.7
StableGM	**86.1** ± **0.9**	**67.9** ± **1.1**	57 ± 0.6

Table 4: Node correctness (%) results on the PPI dataset. The best results are in bold. The second best results are underlined.

ings suggest that our method performs well under low noise settings but struggles with higher levels of noise. Similarly, when it comes to the image datasets, our method shows superior or comparable performance for most classes, but its performance on poorly annotated classes of Pascal VOC Keypoint (*sofa* and *table*) is considerably worse than some of the baselines. Thus, improving the proposed method's performance in settings with high noise or poor/ambiguous annotations could be an area for future research.

We have further carried out experiments to demonstrate the effect of selecting the hardness attention parameter β 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 o 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

Leonid Chindelevitch, Cheng-Yu Ma, Chung-Shou Liao, and Bonnie Berger. Optimizing a global alignment of protein interaction networks. *Bioinformatics*, 29(21):2765–2773, 2013.

Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. Reweighted random walks for graph matching. In *European conference on Computer vision*, pages 492–505. Springer, 2010.

Minsu Cho, Karteek Alahari, and Jean Ponce. Learning graphs to match. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 25–32, 2013.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. A (sub) graph isomorphism algorithm for matching large graphs. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1367–1372, 2004.

Timothee Cour, Praveen Srinivasan, and Jianbo Shi. Balanced graph matching. In *Advances in neural information processing systems*, pages 313–320, 2007.

Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 869–877, 2018.

Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020.

Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d'Aspremont. Convex relaxations for permutation problems. *Advances in neural information processing systems*, 26, 2013.

David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

Quankai Gao, Fudong Wang, Nan Xue, Jin-Gang Yu, and Gui-Song Xia. Deep graph matching under quadratic constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5069–5078, 2021.

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

Dan Gusfield and Robert W Irving. *The stable marriage problem: structure and algorithms*. MIT press, 1989.

Somaye Hashemifar and Jinbo Xu. Hubalign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, 30(17):i438–i444, 2014.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Zheheng Jiang, Hossein Rahmani, Plamen Angelov, Sue Black, and Bryan M Williams. Graph-context attention networks for size-varied deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2343–2352, 2022.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673, 2020.

Oleksii Kuchaiev and Nataša Pržulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Eugene L Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963.

Linfeng Liu, Michael C Hughes, Soha Hassoun, and Liping Liu. Stochastic iterative graph matching. In *International Conference on Machine Learning*, pages 6815–6825. PMLR, 2021.

Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

David G McVitie and Leslie B Wilson. Stable marriage assignment for unequal sets. *BIT Numerical Mathematics*, 10(3):295–309, 1970.

Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019.

Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

Behnam Neyshabur, Ahmadreza Khadem, Somaye Hashemifar, and Seyed Shahriar Arab. Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13): 1654–1662, 2013.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28 (23):3105–3114, 2012.

Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. Deep graph matching via blackbox differentiation of combinatorial solvers. In *European Conference on Computer Vision*, pages 407–424. Springer, 2020.

Ali Saad-Eldin, Benjamin D Pedigo, Carey E Priebe, and Joshua T Vogelstein. Graph matching via optimal transport. *arXiv preprint arXiv:2111.05366*, 2021.

Vikram Saraph and Tijana Milenković. Magna: maximizing accuracy in global network alignment. *Bioinformatics*, 30(20):2931–2940, 2014.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Roded Sharan and Trey Ideker. Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4):427–433, 2006.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Mario Vento and Pasquale Foggia. Graph matching techniques for computer vision. In *Image Processing: Concepts, Methodologies, Tools, and Applications*, pages 381–421. IGI Global, 2013.

Vipin Vijayan, Vikram Saraph, and Tijana Milenković. Magna++: maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, 31(14):2409–2411, 2015.

Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. *arXiv preprint arXiv:1912.02175*, 2019.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

Runzhong Wang, Junchi Yan, and Xiaokang Yang. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3056–3065, 2019.

Runzhong Wang, Junchi Yan, and Xiaokang Yang. Combinatorial learning of robust deep graph matching: an embedding based approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020a.

Runzhong Wang, Junchi Yan, and Xiaokang Yang. 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*, 2021.

Tao Wang, He Liu, Yidong Li, Yi Jin, Xiaohui Hou, and Haibin Ling. Learning combinatorial solver for graph matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7568–7577, 2020b.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32, 2019a.

Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pages 6932–6941. PMLR, 2019b.

Lanling Xu, Jianxun Lian, Wayne Xin Zhao, Ming Gong, Linjun Shou, Daxin Jiang, Xing Xie, and Ji-Rong Wen. Negative sampling for contrastive representation learning: A review. *arXiv preprint arXiv:2206.00212*, 2022.

Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. Learning deep graph matching with channel-independent embedding and hungarian attention. In *International conference on learning representations*, 2020.

Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2684–2693, 2018.

Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2008.

Wen Zhang, Kai Shu, Huan Liu, and Yalin Wang. Graph neural networks for user identity linkage. *arXiv preprint arXiv:1903.02174*, 2019.