

# RoSA: A Robust Self-Aligned Framework for Node-Node Graph Contrastive Learning

Yun Zhu\*, Jianhao Guo\*, Fei Wu and Siliang Tang†

Zhejiang University

{zhuyun\_dcd, guojianhao, wufei, siliang}@zju.edu.cn

## Abstract

Graph contrastive learning has gained significant progress recently. However, existing works have rarely explored non-aligned node-node contrasting. In this paper, we propose a novel graph contrastive learning method named RoSA that focuses on utilizing non-aligned augmented views for node-level representation learning. First, we leverage the earth mover’s distance to model the minimum effort to transform the distribution of one view to the other as our contrastive objective, which does not require alignment between views. Then we introduce adversarial training as an auxiliary method to increase sampling diversity and enhance the robustness of our model. Experimental results show that RoSA outperforms a series of graph contrastive learning frameworks on homophilous, non-homophilous and dynamic graphs, which validates the effectiveness of our work. To the best of our awareness, RoSA is the first work focuses on the non-aligned node-node graph contrastive learning problem. Our codes are available at: <https://github.com/ZhuYun97/RoSA>

## 1 Introduction

Graph representation learning, which aims to learn low dimension representations of nodes and edges for downstream tasks, has become a popular method when dealing with graph-domain data recently. Among all these methods, unsupervised graph contrastive learning has received considerable research attention. It combines the new research trend of graph neural network (GNN) [Kipf and Welling, 2017] and contrastive self-supervised learning [Oord *et al.*, 2018; Chen *et al.*, 2020; Grill *et al.*, 2020] methods, and has achieved promising results on many graph-based tasks [Zhu *et al.*, 2020c; Velickovic *et al.*, 2019; You *et al.*, 2020].

Contrastive learning aims to maximize the agreement between jointly sampled positive views and draw apart the distance between negative views, where in graph domain

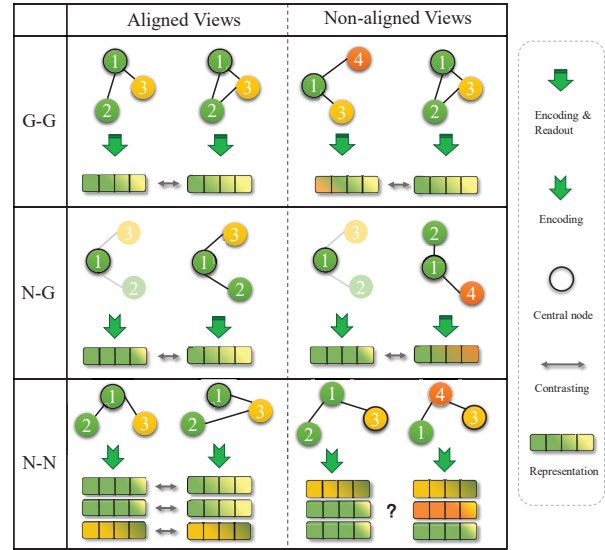


Figure 1: An illustration of different levels of contrasting methods, where G-G means graph-graph, N-G means node-graph and N-N means node-node contrasting level. We only show how a positive pair looks like, where the central node of subgraph is surrounded by a black circle. The number on nodes corresponds to their indices in the original full graph, and the color represents their labels.

we refer augmented subgraph as a “view”. Based on the scale of two contrasted views, graph contrasting learning can be classified as *node-node*, *node-graph*, and *graph-graph* level [Wu *et al.*, 2021]. From another perspective, a pair of contrasted views is recognized as *aligned* or *unaligned* depending on the difference of their node sets. Two aligned views must have identical node indices, except the structure and some features may differ, and two unaligned views can have different node sets. Figure 1 gives an illustrative overview according to this taxonomy.

[Zhu *et al.*, 2021a] indicates that for node-level tasks such as node classification, applying node-node contrasting can obtain the best performance gain. However, previous work for node-node graph contrastive learning all contrast nodes in the aligned scenario which may hinder the flexibility and variability of sampled views and restrict the expressive power of contrastive learning. Moreover, there exist certain circum-

\*Equal Contribution

†Corresponding Author

stances where aligned views are unavailable, for instance the dynamic graphs where the nodes may appear/disappear as time goes by, and the random walk sampling where the views are naturally non-aligned. Compared with aligned node-node contrasting, the non-aligned scenario is able to sample different nodes and their relations more freely, which will assist the model in learning more representative and robust features.

However, applying non-aligned node-node contrasting faces three challenges. First, how to design sub-sampling methods that can generate unaligned views while maintaining semantic consistency? Second, how to contrast two non-aligned views even the number of nodes and correspondence between nodes are inconsistent? Third, how to boost the performance meanwhile enhance the robustness of model for unsupervised graph contrastive learning? None of them have been satisfactorily answered by previous work.

To tackle the challenges discussed above, we propose RoSA: a **R**obust **S**elf-**A**ligned framework for node-node graph contrastive learning. Firstly, we utilize random walk sampling to obtain augmented views for non-aligned node-node contrastive learning. Specifically, for a given graph, we sample a series of subgraphs based on a central node, and two different views of the same central node are treated as a positive pair, while views across different central nodes are selected as negative pairs. Note that even positive pairs are not necessarily aligned. Secondly, inspired by the message passing mechanism of graph neural networks, the node representation can be interpreted as the result of distribution transformation of its neighboring nodes. Intuitively, for a pair of views, we leverage the earth mover’s distance (EMD) to model the minimum effort to transform the distribution of one view to the other as our objective, which can implicitly align different views and capture the changes in their distributions. Thirdly, we introduce unsupervised adversarial training that explicitly operates on node features to increase the diversity of samples and enhance the robustness of our model. To the best of our knowledge, this is the first work that fills the blank in non-aligned node-node graph contrastive learning.

Our main contributions are summarized as follows:

- We propose a robust self-aligned contrastive learning framework for node-node graph representation learning named RoSA. To the best of our knowledge, this is the first work dedicated to solving non-aligned node-node graph contrastive learning problems.
- To tackle the non-aligned problem, we introduce a novel graph-based optimal transport algorithm,  $g$ -EMD, which does not require explicit node-node correspondence and can fully utilize graph topological and attributive information for non-aligned node-node contrasting. Moreover, to compensate for the possible information loss caused by non-aligned sub-sampling, we propose a non-trivial unsupervised graph adversarial training to improve the diversity of sub-sampling and strengthen the robustness of the model.
- Extensive experimental results on various graph settings achieve promising results and outperform several baseline methods by a large margin, which validates the effectiveness and generality of our method.

## 2 Related Works

### 2.1 Self-Supervised Graph Representation Learning

First appeared in the field of computer vision [Oord *et al.*, 2018; He *et al.*, 2020; Grill *et al.*, 2020] and natural language processing [Gao *et al.*, 2021], self-supervised learning showed promising performance in various tasks and applying it to graph domain quickly became a research hot-spot. GraphCL [You *et al.*, 2020] uses different augmentations and applies a readout function to obtain graph-graph level representations, then optimizes the InfoNCE loss, which can be mathematically proved to be the lower bound of mutual information. Inspired by Deep InfoMax [Hjelm *et al.*, 2019], DGI [Velickovic *et al.*, 2019] maximizes the mutual information between patch and graph representations, which is node-graph level contrasting. Recently, node-node level methods like GMI [Peng *et al.*, 2020], GRACE [Zhu *et al.*, 2020c], GCA [Zhu *et al.*, 2021b] and BGRL [Thakoor *et al.*, 2021] show superior performance on node classification task. Unlike DGI, GMI removes the readout function and maximizes the MI between inputs and outputs of the encoder at the node-node level. With graph augmentation methods, GRACE focuses on contrasting aligned views using different nodes as the negative pairs, and the same nodes from different views are regarded as positive pairs, where each positive pair should be aligned first. GCA is similar to GRACE but with adaptive data augmentation. BGRL is a negative-sample-free method which borrows the idea from BGRL [Grill *et al.*, 2020].

Previous works that involve graph level contrasting, usually have a readout function to obtain whole graph representation, which are naturally aligned, but when it comes to node-node level contrasting, they always explicitly align nodes for positive pairs. The work of non-aligned node-node graph contrastive learning has not yet been explored.

### 2.2 Adversarial Training

Adversarial training (AT) has been found useful to improve the model’s robustness. AT is a min-max training process, which aims to maintain the consistency of the model’s output before and after adding adversarial perturbations. Previous works solve the adversarial perturbations from many different perspectives. [Goodfellow *et al.*, 2015] gives a linear approximation of the perturbation under L2 norm (*i.e.* Fast Gradient method). Projected Gradient Descent method [Madry *et al.*, 2018] tries to obtain a more precise perturbation in an iterative manner, but it takes more time, [Shafahi *et al.*, 2019; Zhu *et al.*, 2020a] provide more efficient methods. Lately, [Kong *et al.*, 2020] adopts these methods into the graph domain in a supervised manner. However, unsupervised adversarial training for graphs is still unexplored. In this paper, we adopt AT into our contrastive method to improve the robustness of the model in an unsupervised manner.

## 3 Method

In this chapter, we will introduce the framework of RoSA. Figure 2 gives an overview of RoSA.

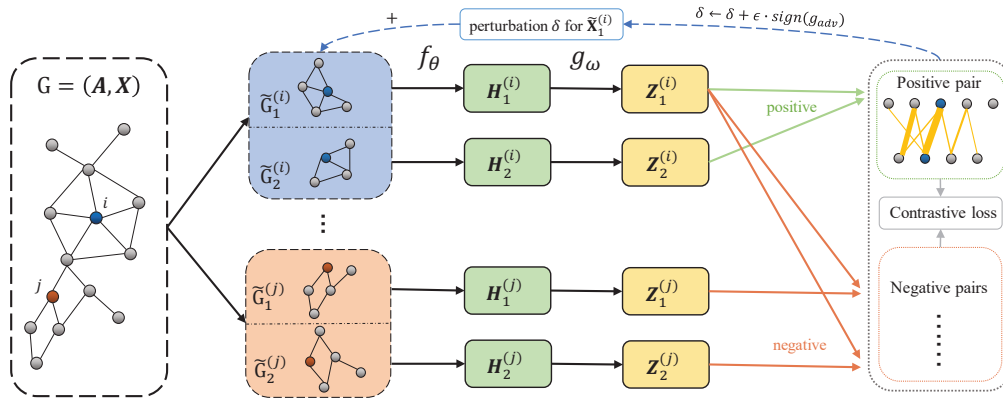


Figure 2: The overview of our proposed method: RoSA. The input is a series of subgraphs sampled from a full graph, where different random walk views from the same central node are recognized as positive pairs and views from different central nodes are treated as negative pairs. Then the subgraphs are fed into the encoder and projector to obtain node embeddings for contrasting. The self-aligned EMD-based contrastive loss will maximize the mutual information (MI) between positive pairs and minimize MI between negative pairs, guiding the model to learn rich representations. Besides, introducing adversarial training into this workflow enhances the robustness of the model.

### 3.1 Preliminaries

Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the set of  $N$  nodes and  $\mathcal{E}$  is the set of  $M$  edges. Also use  $\mathcal{G} = (\mathbf{X}, \mathbf{A})$  to represent graph features, where  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$  represents node feature matrix, each node's feature dimension is  $d$  and can be formulated as  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $\mathbf{A} \in \mathbb{R}^{N \times N}$  represents the graph adjacency matrix, where  $\mathbf{A}_{i,j} = 1$  if an edge exists between node  $i$  and  $j$ , else  $\mathbf{A}_{i,j} = 0$ . For subgraph sampling, each node  $i$  will be treated as central node to get subgraph  $\mathcal{G}^{(i)}$ . An augmented view of subgraph  $\mathcal{G}^{(i)}$  is represented as  $\tilde{\mathcal{G}}_k^{(i)}$  where subscript  $k$  denotes the  $k$ -th augmented view.

### 3.2 Non-Aligned Node-Node Level Sub-Sampling

It has been proven that well-designed data augmentation plays a vital role in boosting the performance of contrastive learning [You *et al.*, 2020]. However, different from the CV and NLP domain, where data is organized in a Euclidean fashion, graph data augmentation methods need to be re-designed and carefully selected.

It is worth noting that in this work, for a positive pair, we need to get different sets of nodes while preserving the consistency of their semantic meanings. Based on such a premise, we propose to utilize random walk with restart sampling as an augmentation method that selects nodes randomly and generates unaligned views. Specifically, random walk sampling starts from the central node  $v$  and generates a random path with a given step size  $s$ . Besides, at each step the walk returns to central node  $v$  with a restart probability  $\alpha$ . The step size  $s$  should not be too large because we want to capture the local structure of the central node. Lastly, edge dropping and feature masking [Zhu *et al.*, 2020c] are applied on subgraphs.

### 3.3 g-EMD: A Self-aligned Contrastive Objective

After obtaining two unaligned augmented views, we define a contrastive objective that measures the agreement of two different views. Prior arts mostly use cosine similarity as a metric to evaluate how far two feature vectors drift apart.

While under our setting, two views may have different and unaligned nodes, where a simple cosine similarity loses its availability. Hence we propose to leverage the earth mover's distance (EMD) as our similarity measure.

EMD [Rubner *et al.*, 2000; Zhang *et al.*, 2020; Liu *et al.*, 2020] is the measure of the distance between two discrete distributions, it can be interpreted as the minimum cost to move one pile of dirt to the other. Although prior work has introduced EMD to the CV domain, the adaptation in the graph domain has not been explored yet. Moreover, according to the characteristics of graph data, we also take topology distance into consideration while computing the cost matrix. Through a non-trivial solution, we combine the vanilla cost matrix and topology distance to obtain a rectified cost matrix which makes the cost related to the node similarity and the distance in the graph topology.

The calculation of  $g$ -EMD can be formulated as a linear optimization problem. In our case, the two augmented views have feature maps  $\mathbf{X} \in \mathbb{R}^{M \times d}$  and  $\mathbf{Y} \in \mathbb{R}^{N \times d}$  respectively, the goal is to measure the distance to transform  $\mathbf{X}$  to  $\mathbf{Y}$ . Suppose for each node  $\mathbf{x}_i \in \mathbb{R}^d$ , it has  $t_i$  units to transport, and node  $\mathbf{y}_j \in \mathbb{R}^d$  has  $r_j$  units to receive. For a given pair of nodes  $\mathbf{x}_i$  and  $\mathbf{y}_j$ , the cost of transportation per unit is  $\mathbf{D}_{ij}$ , and the amount of transportation is  $\Gamma_{ij}$ . With above notations, we can define the linear optimization problem as follows:

$$\begin{aligned} \min_{\Gamma} \quad & \sum_i^M \sum_j^N \mathbf{D}_{ij} \Gamma_{ij}, \\ \text{s.t.} \quad & \Gamma_{ij} \geq 0, i = 1, 2, \dots, M, j = 1, 2, \dots, N \\ & \sum_i^M \Gamma_{ij} = r_j, j = 1, 2, \dots, N \\ & \sum_j^N \Gamma_{ij} = t_i, i = 1, 2, \dots, M \end{aligned} \quad (1)$$

where  $\mathbf{t} \in \mathbb{R}^M$  and  $\mathbf{r} \in \mathbb{R}^N$  are marginal weights for  $\Gamma$  respectively.

The set of all possible transportation matrices  $\Gamma$  can be defined as

$$\Pi(\mathbf{t}, \mathbf{r}) = \{\Gamma \in \mathbb{R}^{M \times N} \mid \Gamma \mathbf{1}_M = \mathbf{t}, \Gamma^T \mathbf{1}_N = \mathbf{r}\}, \quad (2)$$

where  $\mathbf{1}$  is all-one vector with corresponding size, and  $\Pi(\mathbf{t}, \mathbf{r})$  is the set of all possible distributions whose marginal weights are  $\mathbf{t}$  and  $\mathbf{r}$ .

And the cost to transfer  $\mathbf{x}_i$  to  $\mathbf{y}_j$  is defined as

$$\mathbf{D}_{ij} = 1 - \frac{\mathbf{x}_i^T \mathbf{y}_j}{\|\mathbf{x}_i\| \|\mathbf{y}_j\|}, \quad (3)$$

which indicates that nodes with similar representations prefer to generate fewer matching cost between each other. In addition to directly using node representations dissimilarity matrix as a distance matrix, we also take the topology distance  $\Psi \in \mathbb{R}^{M \times N}$  (the smallest hop count between each pair of nodes) into consideration. Nodes are close in topology structure which indicates they may contain similar semantic information. How to combine the representation dissimilarity matrix and topology distance is not a trivial problem. In order not to adjust the original cost matrix  $\mathbf{D}$  drastically, we adopt sigmoid function  $S$  with temperature on topology distance to get re-scale factors  $\mathbf{S} \in [0.5, 1]^{M \times N}$ :

$$\mathbf{S}_{i,j} = S(\Psi_{i,j}) = \frac{1}{1 + e^{-\Psi_{i,j}/\tau}}, \quad (4)$$

where  $\tau \geq 1$  is the temperature factor to control the rate of curve change. We set  $\tau$  as 2 empirically, and leave the choice of different of re-scale function and the tuning of different temperature factors in future work. With the re-scale factors  $\mathbf{S}$ , we can update the cost matrix by

$$\mathbf{D} = \mathbf{D} \circ \mathbf{S}, \quad (5)$$

where  $\circ$  is Hadamard product. In this way, we combine both topology distance and node representation dissimilarity matrix into distance matrix.

As  $\mathbf{D}$  is fixed according to distributions  $\mathbf{X}$ ,  $\mathbf{Y}$  and topology distance, to get g-EMD we need to find the optimal  $\tilde{\Gamma}$ . To solve the optimal  $\tilde{\Gamma}$ , we utilize *Sinkhorn Algorithm* [Cuturi, 2013] by introducing a regularization term:

$$\text{g-EMD}(\mathbf{X}, \mathbf{Y}, \mathbf{S}) = \inf_{\Gamma \in \Pi} \langle \Gamma, \mathbf{D} \rangle_{\text{F}} + \underbrace{\frac{1}{\lambda} \Gamma (\log \Gamma - 1)}_{\text{regularization term}}, \quad (6)$$

where  $\langle \cdot, \cdot \rangle_{\text{F}}$  denotes Frobenius inner product, and  $\lambda$  is a hyper-parameter that controls the strength of regularization. With this regularization, the optimal  $\tilde{\Gamma}$  can be approximated as:

$$\tilde{\Gamma} = \text{diag}(\mathbf{v}) \mathbf{P} \text{diag}(\mathbf{u}), \quad (7)$$

where  $\mathbf{P} = e^{-\lambda \mathbf{D}}$ , and  $\mathbf{v}$ ,  $\mathbf{u}$  are two coefficient vectors whose values can be iteratively updated as

$$\begin{aligned} \mathbf{v}_i^{t+1} &= \frac{\mathbf{t}_i}{\sum_{j=1}^N \mathbf{P}_{ij} \mathbf{u}_j^t}, \\ \mathbf{u}_j^{t+1} &= \frac{\mathbf{r}_j}{\sum_{i=1}^M \mathbf{P}_{ij} \mathbf{v}_i^{t+1}}. \end{aligned} \quad (8)$$

Then the question lies in how to get marginal weights  $\mathbf{t}$  and  $\mathbf{r}$ . The weight represents a node's contribution in comparison of two views, where a node should have larger weight if its semantic meaning is close to the other view. Based on this hypothesis, we define the node weight as dot product between its feature and the mean pooling feature from the other set:

$$\begin{aligned} \mathbf{t}_i &= \max\left\{\mathbf{x}_i^T \cdot \frac{\sum_{j=1}^N \mathbf{y}_j}{N}, 0\right\}, \\ \mathbf{r}_j &= \max\left\{\mathbf{y}_j^T \cdot \frac{\sum_{i=1}^M \mathbf{x}_i}{M}, 0\right\}. \end{aligned} \quad (9)$$

where *max* is to make sure all weights are non-negative, and then both views will be normalized to ensure having the same amount of features to transport.

With optimal transportation amount  $\tilde{\Gamma}$ , we obtain:

$$\text{g-EMD}(\mathbf{X}, \mathbf{Y}, \mathbf{S}) = \langle \tilde{\Gamma}, \mathbf{D} \rangle_{\text{F}}. \quad (10)$$

Now we can leverage EMD as the distance measure to contrastive loss objective. For any central node  $\mathbf{v}_i$  and its augmented graph views  $(\tilde{\mathcal{G}}_1^{(i)}, \tilde{\mathcal{G}}_2^{(i)})$ , an encoder  $f_\theta$  (e.g. GNN) is applied to get embeddings  $\mathbf{H}_1^{(i)}$  and  $\mathbf{H}_2^{(i)}$  respectively, then a linear projector  $g_\omega$  is applied on top of that to get  $\mathbf{Z}_1^{(i)}$  and  $\mathbf{Z}_2^{(i)}$  to improve generality for downstream tasks as indicated in [Chen *et al.*, 2020]. Formally, we define the EMD-based contrastive loss for node  $\mathbf{v}_i$  as

$$\begin{aligned} \ell(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(i)}) &= \\ -\log\left(\frac{e^{s(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(i)})/\tau}}{\sum_{k=1}^N e^{s(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(k)})/\tau} + \sum_{k=1}^N \mathbf{1}_{[k \neq i]} e^{s(\mathbf{Z}_1^{(i)}, \mathbf{Z}_1^{(k)})/\tau}}\right), \end{aligned} \quad (11)$$

where  $s(\mathbf{x}, \mathbf{y})$  is a function that calculates the similarity between  $\mathbf{x}$  and  $\mathbf{y}$ , here we use  $1 - \text{EMD}(\mathbf{x}, \mathbf{y})$  to replace  $s(\mathbf{x}, \mathbf{y})$ ;  $\mathbf{1}$  is an indicator function which returns 1 if  $i \neq k$  otherwise returns 0; and  $\tau$  is temperature parameter. Adding all nodes in  $\mathcal{N}$ , the overall contrastive loss is given by:

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^N \left[ \ell(\mathbf{Z}_1^{(i)}, \mathbf{Z}_2^{(i)}) + \ell(\mathbf{Z}_2^{(i)}, \mathbf{Z}_1^{(i)}) \right]. \quad (12)$$

We summarize our proposed algorithm for non-aligned node-node contrastive learning in Appendix A.

### 3.4 Unsupervised Adversarial Training

Adversarial training can be considered as an augmentation technique which aims to improve the model's robustness. [Kong *et al.*, 2020] has empirically proven that graph adversarial augmentation on feature space can boost the performance of GNN under a supervised manner. Such a method can be modified for graph contrastive learning as

$$\min_{\theta, \omega} \mathbb{E}_{(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}) \sim \mathbb{D}} \left[ \frac{1}{M} \sum_{t=0}^{M-1} \max_{\delta_t \in \mathcal{I}_t} \mathcal{J}(\mathbf{x}_1^{(i)} + \delta_t, \mathbf{x}_2^{(i)}) \right], \quad (13)$$

where  $\theta, \omega$  are the parameters of encoder and projector,  $\mathbb{D}$  is data distribution,  $\mathcal{I}_t = \mathcal{B}_{\mathbf{X}+\delta_0}(\alpha t) \cap \mathcal{B}_{\mathbf{X}}(\epsilon)$  where  $\epsilon$  is the perturbation budget. For efficiency, the inner loop runs  $M$  times, the gradient of  $\delta, \theta_{t-1}$  and  $\omega_{t-1}$  will be accumulated in each time, and the accumulated gradients will be used for updating  $\theta_{t-1}$  and  $\omega_{t-1}$  during outer update. Equipped with such adversarial augmentation, we complete a more robust self-aligned task. The energy is hopefully transferred between nodes belonging to different categories during max-process, and min-process will remedy such a bad situation to make the alignment more robust. In this way, the adversarial augmentation increases the diversity of samples and improves the robustness of the model.

## 4 Experiments

We conduct extensive experiments on ten public benchmark datasets to evaluate the effectiveness of RoSA. We use RoSA to learn node representations in an unsupervised manner and assess their quality by a linear classifier trained on top of that. Some more detailed information about datasets and experimental setup can be found in Appendix B, C.

### 4.1 Datasets

We conduct experiments on ten public benchmark datasets that include four homophilous datasets (Cora, Citeseer, Pubmed and DBLP), three non-homophilous datasets (Cornell, Wisconsin and Texas), two large-scale inductive datasets (Flickr and Reddit) and one dynamic graph dataset (CIAW) to evaluate the effectiveness of RoSA. Details of datasets can be found in Appendix B.

### 4.2 Experimental Setup

**Models** For small-scale datasets, we apply a two-layer GCN as our encoder  $f_\theta$  and for the large-scale datasets (Flickr and Reddit), we adopt a three-layer GraphSAGE-GCN [Hamilton *et al.*, 2017] with residual connections as the encoder following DGI [Velickovic *et al.*, 2019] and GRACE [Zhu *et al.*, 2020c]. The formulas of encoders can be found in Appendix C. Specifically, similar to [Chen *et al.*, 2020], a projection head which comprises a two-layer non-linear MLP with BN is added on top of the encoder. Detailed hyperparameter settings are in Appendix C.

**Baselines** We compare RoSA with two node-graph constrasting methods DGI [Velickovic *et al.*, 2019], SUBG-CON [Jiao *et al.*, 2020]), and four node-node methods GMI [Peng *et al.*, 2020], GRACE [Zhu *et al.*, 2020c], GCA [Zhu *et al.*, 2021b] and BGRL [Thakoor *et al.*, 2021].

### 4.3 Results and Analysis

**Results for homophilous datasets** Table 1 shows the node classification results on four homophilous datasets, some of the reported statistics are borrowed from [Zhu *et al.*, 2020c]. Experiment results show that N-N methods surpass N-G on node classification tasks. And RoSA is superior to all baselines and achieves state-of-the-art performance, and even surpasses the supervised method (GCN), which proves the effectiveness of leveraging EMD-based contrastive loss and adversarial training in non-aligned node-node scenarios. Different from other node-node methods that train on full graphs,

Method	Level	Cora	Citeseer	Pubmed	DBLP
Raw Features	-	64.8	64.6	84.8	71.6
DeepWalk	-	67.2	43.2	65.3	75.9
GCN	-	82.8	72.0	84.9	82.7
DGI	N-G	82.6±0.4	68.8±0.7	86.0±0.1	83.2±0.1
SUBG-CON*	N-G	82.6±0.9	69.2±1.3	84.3±0.3	83.8±0.3
GMI	N-N	82.9±1.1	70.4±0.6	84.8±0.4	84.1±0.2
GRACE	N-N	83.3±0.4	72.1±0.5	86.7±0.1	84.2±0.1
GCA	N-N	83.8±0.8	72.2±0.7	86.9±0.2	84.3±0.2
BGRL	N-N	83.8±1.6	72.3±0.9	86.0±0.3	84.1±0.2
RoSA	N-N	<b>84.5±0.8</b>	<b>73.4±0.5</b>	<b>87.1±0.2</b>	<b>85.0±0.2</b>

Table 1: Summary of classification accuracy of node classification tasks on homophilous graphs. The second column represents the contrasting mode of methods, N-G stands for node-graph level, and N-N stands for node-node level. For a fair comparison, in SUBG-CON\* we replace the original encoder with the encoder used in our paper and apply the same evaluation protocol as ours.

our method is trained on various non-aligned subgraphs, which brings more flexibility but also non-alignment challenge. RoSA learns more information from the challenging pretext task. The visualization of cost matrix and transportation matrix in EMD during training is in Appendix E.

Methods	Cornell	Wiscons.	Texas	Cornell	Wiscons.	Texas
DGI	56.3±4.7	50.9±5.5	56.9±6.3	58.1±4.1	52.1±6.3	57.8±5.2
SUBG-CON	54.1±6.7	48.3±4.8	56.9±6.9	58.7±6.8	59.0±7.8	61.1±7.3
GMI	58.1±4.0	52.9±4.2	57.8±5.9	69.6±5.3	70.8±5.2	69.6±5.3
GRACE	58.2±4.1	54.3±7.1	58.9±4.7	72.3±5.3	74.1±5.5	69.8±7.2
RoSA	<b>59.3±3.6</b>	<b>55.1±4.7</b>	<b>60.3±4.5</b>	<b>74.3±6.2</b>	<b>77.1±4.3</b>	<b>71.1±6.6</b>

Table 2: Non-homophilous node classification using GCN (left) and MLP (right).

**Results for non-homophilous dataset** Previous works have shown that GCN performs poorly on non-homophilous graphs [Pei *et al.*, 2020; Zhu *et al.*, 2020b], because there are a lot of high-frequency signals on such graphs, and GCN is essentially a low-pass filter, where a lot of useful information will be filtered out. Since the design of the encoder is not the focus of our work, we use both GCN and MLP as our encoders in this part.

We compare the performance of our model with DGI, SUBG-CON, GMI, GRACE using either GCN or MLP as encoder, see Table 2. From the statistics, we can summarize three major conclusions: Firstly, the overall performance of SUBG-CON and DGI lags behind the others. This is because SUBG-CON and DGI are the node-graph level contrasting methods that maximize the mutual information between central node representation and its contextual subgraph representation, and under the non-homophilous circumstance, the contextual graph representation gathers highly variant features from different kinds of nodes, which renders wrong and meaningless signals.

Secondly, with the same method, the MLP version performs significantly better than its GCN counterpart, which confirms the statement that MLP is more suitable for non-homophilous graphs. Furthermore, we can observe that the performance gap between node-global and node-node methods widens when using MLP as the encoder. We suspect such a phenomenon is caused because the GCN encoder loses a

large amount of information under a non-homophilous setting and makes the effort of other modules in vain.

Thirdly and most importantly, RoSA outperforms other benchmarks on all three datasets, no matter the choice of the encoder, which validates the effectiveness of RoSA for non-homophilous graphs. We speculate that RoSA will tighten the distance of nodes of the same class.

**Result for inductive learning on large-scale datasets** The experiments conducted above are all under the transductive setting. In this part, the experiments are under the inductive setting where tests are conducted on unseen or untrained nodes. The micro-averaged F1 score is used for both of these two datasets. The results are shown in Table 3, we can see that RoSA works well on large-scale graphs under inductive setting and reaches state-of-the-art performance. An explanation is DGI, GMI and GRACE can not directly work on full graphs, they use the sampling strategy proposed by [Hamilton *et al.*, 2017] in their original work. However, we adopt subsampling (random walk) as our augmentation technique which means our method can seamlessly work on these large graphs. Furthermore, our pretext task is designed for such subsampling which is more suitable for large graphs.

Methods	Flickr	Reddit
Raw features	20.3	58.5
DeepWalk	27.9	32.4
FastGCN	48.1±0.5	89.5±1.2
GraphSAGE	50.1±1.3	92.1±1.1
Unsup-GraphSAGE	36.5	90.8
DGI	42.9±0.1	94.0±0.1
GMI	44.5±0.2	95.0±0.0
GRACE	48.0±0.1	94.2±0.0
RoSA	<b>51.2±0.1</b>	<b>95.2±0.0</b>

Table 3: Result for inductive learning on large-scale datasets.

**Results for dynamic graphs dataset** In addition, we also test our method on dynamic graphs. For the contrastive task, we consider the adjacent snapshots as positive views because the evolution process is generally "smooth", and the snapshots far away from the anchor are considered as negative views. In CIAW, each snapshot maintains all nodes appeared in the timeline, however, in real-world scenarios, the addition or deletion of nodes happens as time goes by. So in CIAW\*, we remove isolated nodes in each snapshot to emulate such a situation. Note that GRACE can not work on CIAW\* because CIAW\* creates a non-aligned situation, while GRACE is inherently an aligned method. From the statistics in Table 4, RoSA surpasses other competitors and can work well in both situations. Currently, we simply use static GNN encoder with discrete-time paradigms which can be replaced with temporal GNN encoders, and we will leave it for future work.

#### 4.4 Ablation Study

To prove the effectiveness of the design of RoSA, we conduct ablation experiments masking different components under the

	CIAW	CIAW*
GraphSAGE	64.0±8.5	69.7±10.1
GRACE	65.3±7.9	-
RoSA	<b>67.6±7.0</b>	<b>73.2±9.3</b>

Table 4: Node classification using GraphSAGE on dynamic graphs.

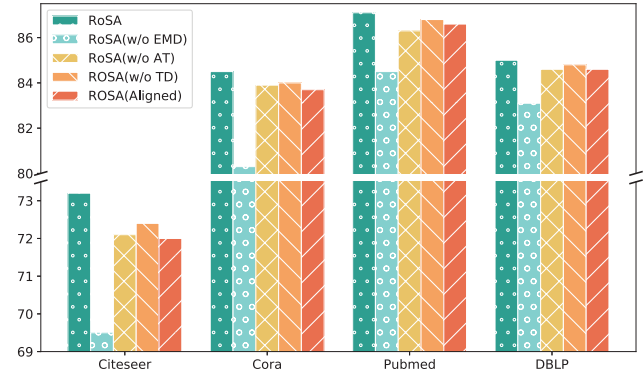


Figure 3: Ablation study on RoSA

same hyperparameters. First we replace the EMD-based InfoNCE loss with a regular cosine similarity metric, represented as RoSA w/o EMD (In order to make it computable under such situation, we restrict the same amount of nodes for contrasted views). Second we use the vanilla cost matrix for EMD, named as RoSA w/o TD. Then we remove the adversarial training process, denoted as RoSA w/o AT. Finally, we adopt aligned views contrasting instead of the original non-aligned random walking, named as RoSA Aligned. For a fair comparison, we keep other hyperparameters and the training scheme same. The results is summarized in Figure 3. As we can see, the performance degrades without either EMD, adversarial training or rectified cost matrix, which indicates the effectiveness of the corresponding components. Furthermore, compared to aligned views, the model achieves comparable or even better results under the non-aligned condition, which demonstrates that our model, to a certain degree, solves the non-aligned graph contrasting problem. The experiments of sensitivity analysis are in Appendix D.

## 5 Conclusion

In this paper, we propose a robust self-aligned framework for node-node graph contrastive learning, where we design and utilize the graph-based earth mover's distance ( $g$ -EMD) as a similarity measure in the contrastive loss to avoid explicit alignment between contrasted views. Then we introduce unsupervised adversarial training into graph domain to further improve the robustness of the model. Extensive experiment results on homophilous, non-homophilous and dynamic graphs datasets demonstrate that our model can effectively be applied to non-aligned situations and outperform other competitors. Moreover, in this work we adopt simple random walk with restart as the subsampling technique, and RoSA may achieve better performance if equipped with more powerful sampling methods in future work.

## References

- [Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. of ICML*, 2020.
- [Ciotti *et al.*, 2016] Valerio Ciotti, Moreno Bonaventura, Vincenzo Nicosia, Pietro Panzarasa, and Vito Latora. Homophily and missing links in citation networks. *EPJ Data Science*, 2016.
- [Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Light-speed computation of optimal transport. In *Proc. of NeurIPS*, 2013.
- [Fey and Lenssen, 2019] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *ArXiv preprint*, 2019.
- [Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv preprint*, 2021.
- [Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *Proc. of AISTATS*, 2010.
- [Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR*, 2015.
- [Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhao-han Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *Proc. of NeurIPS*, 2020.
- [Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proc. of KDD*, 2016.
- [Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proc. of ICONIP*, 2017.
- [He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. of CVPR*, 2020.
- [Hjelm *et al.*, 2019] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proc. of ICLR*, 2019.
- [Jiao *et al.*, 2020] Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyong Zhu. Sub-graph contrast for scalable self-supervised graph representation learning, 2020.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*, 2017.
- [Kong *et al.*, 2020] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Flag: Adversarial data augmentation for graph neural networks. *ArXiv preprint*, 2020.
- [Liu *et al.*, 2020] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *ArXiv preprint*, 2020.
- [Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of ICLR*, 2018.
- [McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.
- [Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv preprint*, 2018.
- [Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of NeurIPS*, 2019.
- [Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 2011.
- [Pei *et al.*, 2020] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *Proc. of ICLR*, 2020.
- [Peng *et al.*, 2020] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph Representation Learning via Graphical Mutual Information Maximization. In *Proc. of WWW*, 2020.
- [Rubner *et al.*, 2000] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover’s distance as a metric for image retrieval. *International journal of computer vision*, 2000.
- [Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008.
- [Shafahi *et al.*, 2019] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Proc. of NeurIPS*, 2019.
- [Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan

- Günemann. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop, NeurIPS 2018*, 2018.
- [Thakoor *et al.*, 2021] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.
- [Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.
- [Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *Proc. of ICLR*, 2019.
- [Wu *et al.*, 2021] Lirong Wu, Haitao Lin, Zhangyang Gao, Cheng Tan, Stan Li, et al. Self-supervised on graphs: Contrastive, generative, or predictive. *ArXiv preprint*, 2021.
- [You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Proc. of NeurIPS*, 2020.
- [Zhang *et al.*, 2020] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover’s distance and structured classifiers. In *Proc. of CVPR*, 2020.
- [Zhu *et al.*, 2020a] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freeb: Enhanced adversarial training for natural language understanding. In *Proc. of ICLR*, 2020.
- [Zhu *et al.*, 2020b] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs, 2020.
- [Zhu *et al.*, 2020c] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning, 2020.
- [Zhu *et al.*, 2021a] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*, 2021.
- [Zhu *et al.*, 2021b] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proc. of WWW*, 2021.



## A. Algorithm

The steps of the procedure of our method are summarised as:

---

**Algorithm 1** Algorithm for a robust self-aligned framework for node-node graph contrastive learning.

---

**Input:** Sampler function  $\mathcal{T}(\mathcal{G}, \text{idx})$ , ascent steps  $M$ , ascent step size  $\alpha$ , additional augmentations  $\tau_\alpha, \tau_\beta$ , encoder  $f_\theta$ , projector  $g_\omega$ , and training graph  $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$

```

1: while not converge do
2:   for  $i = 1$  to  $N$  do
3:     Sample two context subgraphs  $\mathcal{G}_1^{(i)} = \mathcal{T}(\mathcal{G}, i)$ ,
        $\mathcal{G}_2^{(i)} = \mathcal{T}(\mathcal{G}, i)$  for each central node
4:   end for
5:    $\mathbf{S} = \{(\mathcal{G}_1^{(i)}, \mathcal{G}_2^{(i)})\}, i = 1 \dots N$ 
6:   sampled batch  $\mathcal{B} = \{(\mathcal{G}_1^{(k)}, \mathcal{G}_2^{(k)})\} \in \mathbf{S}$ 
7:    $\mathcal{B}_1 = \{\mathcal{G}_1^{(k)}\}, \mathcal{B}_2 = \{\mathcal{G}_2^{(k)}\}$ 
8:    $\{(\tilde{\mathbf{X}}_1^{(k)}, \tilde{\mathbf{A}}_1^{(k)})\} = \tilde{\mathcal{B}}_1 = \tau_\alpha(\mathcal{B}_1)$ 
9:    $\{(\tilde{\mathbf{X}}_2^{(k)}, \tilde{\mathbf{A}}_2^{(k)})\} = \tilde{\mathcal{B}}_2 = \tau_\beta(\mathcal{B}_2)$ 
10:   $\delta_0 \leftarrow U(-\alpha, \alpha)$ 
11:   $g_0 \leftarrow 0$ 
12:  for  $t = 1 \dots M$  do
13:     $\mathbf{Z}_1 = g_\omega \circ f_\theta(\tilde{\mathbf{X}}_1 + \delta_{t-1}, \tilde{\mathbf{A}}_1) = g_\omega \circ f_\theta(\tilde{\mathcal{B}}_1 + \delta_{t-1})$ 
14:     $\mathbf{Z}_2 = g_\omega \circ f_\theta(\tilde{\mathcal{B}}_2)$ 
15:     $\mathbf{g}_t \leftarrow \mathbf{g}_{t-1} + \frac{1}{M} \cdot \nabla_{\theta, \omega} \ell(\mathbf{Z}_1, \mathbf{Z}_2)$ 
16:     $\mathbf{g}_\delta \leftarrow \nabla_{\delta} \ell(\mathbf{Z}_1, \mathbf{Z}_2)$ 
17:     $\delta_t \leftarrow \delta_{t-1} + \alpha \cdot \mathbf{g}_\delta / \|\mathbf{g}_\delta\|_F$ 
18:  end for
19:   $\theta \leftarrow \theta - \tau \cdot \mathbf{g}_{M, \theta}$ 
20:   $\omega \leftarrow \omega - \tau \cdot \mathbf{g}_{M, \omega}$ 
21: end while

```

---

## B. Dataset Details

We will introduce the details of all datasets used in experiments. The statistics of datasets are in Table 6. All the datasets are available on Pytorch Geometry Library (PyG) [Fey and Lenssen, 2019]

**Homophilous datasets** We use four citation network datasets [Sen *et al.*, 2008], Cora, Citeseer, Pubmed and DBLP. As for these datasets, nodes represent a variety of papers, and edges represent citation relationships between these papers. Node features are represented as the bag-of-word model of the corresponding paper, and the label is the academic topic of the paper. These four datasets are highly homophilous graphs that most of edges connect nodes sharing the same labels. Following GRACE, we also randomly split the nodes into (10%/10%/80%) for train/validation/test respectively instead of using the standard fixed splits which are unreliable for evaluating GNN methods [Shchur *et al.*, 2018].

**Non-homophilous datasets** In real-world scenarios, the pattern "like attracts like" exists in many networks (*e.g.*,

friendship networks [McPherson *et al.*, 2001], citation networks [Ciotti *et al.*, 2016]), but there also exists different pattern as "opposites attract" (*e.g.*, dating networks or molecular networks [Zhu *et al.*, 2020b]). We can use edge homophily ratio  $h = \frac{|\{(i,j):(i,j) \in \mathcal{E} \wedge y_i = y_j\}|}{|\mathcal{E}|}$  to indicate the portion of edges that connect two nodes with same labels. A graph is considered to be non-homophilous if  $h < 0.5$ . Numerous GNN variants may fail on non-homophilous datasets because the aggregation functions can be considered as feature smoothing, and feature smoothing will average nodes' features even if they have different labels. In order to verify that our method is more suitable for non-homophilous graphs, we also conduct experiments on three non-homophilous datasets collected by the CMU WebKB project, which are Cornell, Texas, and Wisconsin. These three datasets are webpage datasets collected from science departments of corresponding universities, where nodes represent web pages and edges represent hyperlinks between them. Node features are the bag-of-words representation of web pages, and these nodes are manually classified into five categories: Student, Project, Course, Staff, and Faculty. We use the preprocessed version in [Pei *et al.*, 2020] with the standard dataset split. The detailed information of datasets is summarized in Table 6.

**Inductive learning on large graphs** We use two commonly used large graphs (Flickr and Reddit) to evaluate our method under inductive learning setting. Each node in Flickr represents an uploaded image. Edges are formed between nodes (images) from the same location, submitted to the same gallery, group, or set, images sharing common tags, images taken by friends, etc. Each node contains the 500-dimensional bag-of-word representation of the images provided by NUS-wide\*. And the labels are generated according to the tags of the images. Reddit is a social network with Reddit posts created in September 2014 which is preprocessed by [Hamilton *et al.*, 2017]. In the dataset, each node represents a post, and edges connect posts if the same user has commented on both. Each node contains 602-dimensional off-the-shelf GloVe word embeddings which are constructed from the post title, content, and comments, along with other metrics such as post score and the number of comments. We use the data splits processed by [Hamilton *et al.*, 2017]. Posts in the first 20 days are for training, including 151,708 nodes, and the remaining for testing (with 30% data including 23,699 nodes for validation). The inductive setting follows [Velickovic *et al.*, 2019], validation and test nodes are invisible to the training algorithm.

**Dynamic graph dataset** We use one real-world dataset which is called The Contacts In A Workplace (CIAW)<sup>†</sup>. It is a vertex-focused dataset of [35] that contains the temporal network of contacts between individuals measured in an office building in France, from June 24, to July 3, 2013. Each node represents a worker wearing a sensor which can record the interaction with another worker within 1.5 m. The edges will be constructed between nodes if they have contacts (interaction lasting more than 20 s). For each 20s interval be-

\*<https://lms.comp.nus.edu.sg/research/NUS-WIDE.htm>

<sup>†</sup><http://www.sociopatterns.org/datasets/contacts-in-a-workplace/>

	Hidden size	Batch size	Learning rate	Weight decay	Walk length	Epochs	Patience	Optimizer	$\tau$	$p_{e,1}$	$p_{e,2}$	$p_{f1}$	$p_{f2}$
Cora	128	128	1e-2	5e-4	10	500	-	SGD	0.4	0.2	0.2	0.3	0.3
Citeseer	256	128	1e-2	5e-4	10	300	-	SGD	0.7	0.5	0.4	0.5	0.4
Pubmed	256	256	1e-3	5e-4	10	500	-	AdamW	0.1	0.4	0.1	0.0	0.2
DBLP	256	128	1e-3	5e-4	10	500	-	AdamW	0.8	0.1	0.2	0.2	0.3
Flickr	512	128	1e-3	5e-4	20	200	-	AdamW	0.1	0.0	0.2	0.2	0.2
Reddit	512	128	1e-3	5e-4	20	200	-	AdamW	0.2	0.4	0.1	0.0	0.2
Cornell	64	256	1e-3	5e-4	10	200	20	SGD	0.4	0.2	0.3	0.2	0.3
Wisconsin	64	256	1e-3	5e-4	10	200	20	SGD	0.4	0.2	0.3	0.2	0.3
Texas	64	256	1e-3	5e-4	10	200	20	SGD	0.4	0.2	0.3	0.2	0.3
CIAW	128	9	1e-2	5e-4	-	200	20	SGD	0.4	0.2	0.3	0.2	0.3

Table 5: Hyperparameters specifications

tween June 24, and July 3, 2013, all the contacts occurring between the surveyed individuals (nodes) have been recorded. Each node is further characterized by his or her department name used as their labels. The task is to predict each individual’s department by leveraging the historical sequence of their interactions. For preprocessing CIAW, we downsample this dynamic graph into 20 discrete snapshots according to timestamp. The latest two snapshots are used for testing. Further, we randomly split nodes into 1:9 as training and testing nodes in each individual experiment. Moreover, we apply Node2vec [Grover and Leskovec, 2016] to generate 64-dimensional representation for each node by training graphs. For preprocessing CIAW\*, we remove isolated nodes in each snapshot to simulate the situation of the addition and deletion of the nodes over time. For the training and testing process, in a semi-supervised manner, we will use training graphs to train the encoder through backwarding on labeled training nodes, and then test the performance of the encoder using testing nodes in testing graphs. In an unsupervised manner, we consider the adjacent snapshots as positive views because the evolution process is generally “smooth”. And the snapshots which are far away from the anchor are considered as negative views. Although we simply use static GNN encoder with discrete-time paradigms in our work, it can be applied to temporal GNN encoders with continuous-time paradigms. We leave this in future work.

Dataset	# N	# E	# F	# C	H
Cora	2,078	5,278	1,433	7	0.81
CiteSeer	3,327	4,676	3,703	6	0.74
PubMed	19,717	44,327	500	3	0.80
DBLP	17,716	105,734	1,639	4	0.83
Cornell	183	280	1,703	5	0.30
Texas	183	295	1,703	5	0.11
Wisconsin	251	466	1,703	5	0.21
Flickr	89,250	899,756	500	7	0.32
Reddit	231,443	11,606,919	602	41	0.76
CIAW	92	9,827	64	5	-

Table 6: Details of used datasets, where we substitute N for *Nodes*, E for *Edges*, F for *Features*, C for *Classes*, H for *Homophily ratio*.

## C. Implementation Details

**Model architecture** We use two kinds of GNN encoders following [Zhu *et al.*, 2020c; Velickovic *et al.*, 2019]. On small-scale datasets, we adopt two layer GCN as:

$$\text{GCN}_i(\mathbf{X}, \mathbf{A}) = \sigma \left( \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X} \mathbf{W}_i \right), \quad (14)$$

$$\mathbf{H} = \text{GCN}_2(\text{GCN}_1(\mathbf{X}, \mathbf{A}), \mathbf{A}), \quad (15)$$

where  $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is the adjacency matrix with self loop, and  $\hat{\mathbf{D}}$  is the degree matrix of  $\hat{\mathbf{A}}$ .  $\sigma$  is an activation function,  $\mathbf{W}$  is a trainable linear transformation for input feature  $\mathbf{X}$ .

As for the large-scale datasets (Flickr and Reddit), we adopt a three-layer GraphSAGE-GCN [Hamilton *et al.*, 2017] with residual connections as the encoder following DGI and GRACE:

$$\text{MP}_i(\mathbf{X}, \mathbf{A}) = \hat{\mathbf{D}}^{-1} \hat{\mathbf{A}} \mathbf{X} \mathbf{W}_i \quad (16)$$

$$\widetilde{\text{MP}}_i(\mathbf{X}, \mathbf{A}) = \sigma(\mathbf{X} \mathbf{W}'_i) \text{MP}_i(\mathbf{X}, \mathbf{A}) \quad (17)$$

$$\mathbf{H} = \widetilde{\text{MP}}_3 \left( \widetilde{\text{MP}}_2 \left( \widetilde{\text{MP}}_1(\mathbf{X}, \mathbf{A}), \mathbf{A} \right), \mathbf{A} \right). \quad (18)$$

**Evaluation metrics** We evaluate the learned encoder as follows. Firstly, we train the model in an unsupervised manner. Then, we extract node embeddings using the fixed pre-trained model. Lastly, a linear classifier will be trained on these embeddings across the training set and give the results on the test nodes. For four citation networks, we use an  $l_2$ -regularization LogisticRegression classifier from Scikit-Learn [Pedregosa *et al.*, 2011] using the ‘liblinear’ solver following [Zhu *et al.*, 2020c]. For other datasets, we use one layer MLP through 100 epochs with Adam optimizer. We train the model for 20 runs and report the average classification accuracy or micro-averaged F1 score (on Flickr and Reddit) along with its standard deviation.

**Computer infrastructures specifications** For hardwares, all experiments are conducted on a computer server with eight GeForce RTX 3090 GPUs with 24GB memory and 64 AMD EPYC 7302 CPUs. Besides, our models are implemented by Pytorch Geometric 1.7.0 [Fey and Lenssen, 2019] and Pytorch 1.8.1 [Paszke *et al.*, 2019]. All the datasets used in our work are available in PyTorch Geometric libraries.

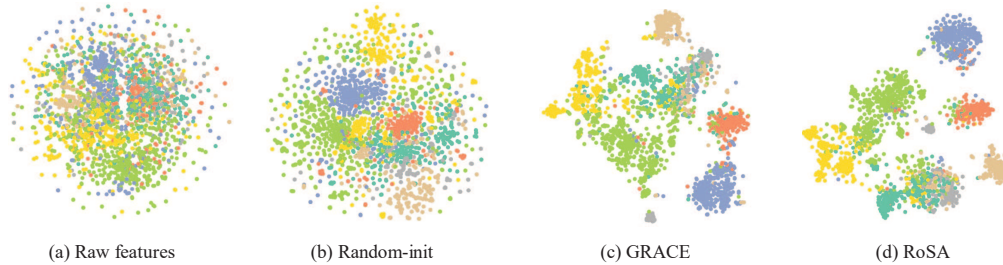


Figure 4: t-SNE visualization of node embeddings on Cora dataset, (a) is the raw features, (b) depicts features from a randomly initialized RoSA model, (c) shows embeddings from trained GRACE model, (d) is the result of trained RoSA. The margins of each cluster learned from RoSA are much wider than the learned GRACE.

**Hyperparameters** All hyperparameters used in experiments are listed in Table 5.  $p_{e,1}, p_{e,2}, p_{f,1}, p_{f,2}$  are the probability parameters that control the extent of data augmentations like GRACE [Zhu *et al.*, 2020c].  $p_{e,1}, p_{e,2}$  is used for controlling the ratio of dropping edges and  $p_{f,1}, p_{f,2}$  decides what a fraction of feature dimensions will be masked. For subsampling, we set the restart ratio as 0.8 on Pubmed and 0.5 on others. For one epoch, we only generate subgraphs for partial central nodes (limited by batch size). All models are initialized with Glorot initialization [Glorot and Bengio, 2010]. During the training process, we use an early stopping strategy on the observed results of the training loss with specific patience.

As for unsupervised adversarial training, we adopt that the inner loop runs 3 times ( $M = 3$ ) and set step size  $\alpha$  as  $10^{-3}$  to implicitly control perturbation budget  $\epsilon$ . The perturbation is not bounded by a definite  $\epsilon$ . The accumulated gradients for model parameters ( $\theta, \omega$ ) during the inner loop will be used in the outer update like [Zhu *et al.*, 2020a; Kong *et al.*, 2020]. In addition, we only add the adversarial perturbation  $\delta$  to one view rather than two augmented views. Concerning the order of augmentations, we firstly use subsampling to obtain a number of subgraphs, then edge dropping and feature masking will be applied on subgraphs. Lastly, an adversarial perturbation will be added to node features to improve model robustness.

Regarding sinkhorn algorithm [Cuturi, 2013], we set the iteration number as 5 for computing transportation matrix  $\mathbf{P}$  with  $\lambda$  equaling to 20 in the regularization term. We find that tuning these two hyperparameters slightly changes performance because the main aim of energy transmutation is not changed.

## D. Additional Experiments

**Sensitivity analysis** Firstly, we explore the influence of different walk length (steps) in sampling process. We measure how the performance is affected by varying walk length in the range of  $\{5, 10, 20, 30, 40, 50\}$ . The results on Cora and Citeseer dataset are depicted in Figure 5. We get comparable results as the walk length reaches 10. After that, as the walk length gets larger, the accuracy drops. We guess that is because larger walk length (bigger subgraph) will introduce more noises. For instance, the negative samples are prone to contain more consistent substructures that can be considered positive signals, thus confusing the model to distinguish be-

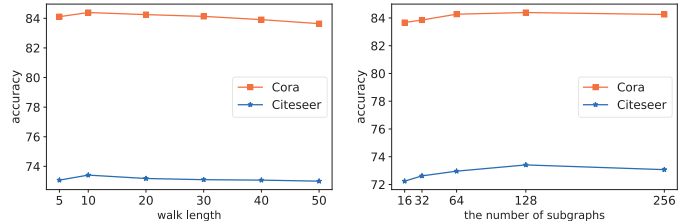


Figure 5: Analysis on critical hyperparameters. The left figure shows the impact of the walk length, the right figure embodies the influence of subgraph number.

tween positive and negative samples.

Secondly, we test the impact of different number of subgraphs trained in each epoch. This factor will determine the amount of negative samples during training. When the number reaches around 64, the accuracy becomes stable.

## E. Visualization

**The visualization of one instance of EMD** We visualize one contrasted pair of Cora in Figure 6. As we can see, with two non-aligned subgraphs, introducing EMD can lead to a pseudo alignment process. More specifically, the distribution transport tends to happen more frequently between nodes with similar semantic meaning, which helps the model learn meaningful representations.

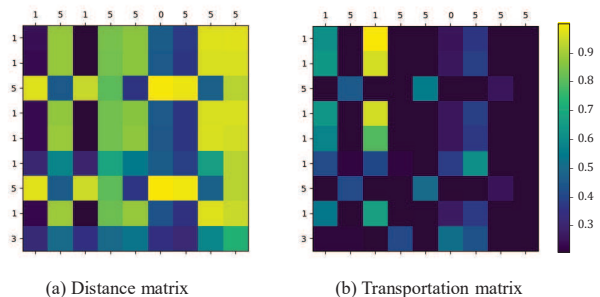


Figure 6: The distance matrix  $\mathbf{D}$  and transportation matrix  $\mathbf{\Gamma}$  of two contrasted views, where grid with high brightness has greater value. The x-axis means the labels of nodes in the first view and y-axis means labels in the second view. The energy transfer mostly occurs between nodes with the same category.

**Embedding visualization** In order to assess the quality of learned embeddings, we adopt t-SNE [Van der Maaten and Hinton, 2008] to visualize the node embedding on Cora dataset using raw features, random-init of RoSA, GRACE, and RoSA , where different classes have different colors in Figure 4. We can observe that the 2D projection of node embeddings learned by RoSA has a clear separation of clusters, which indicates the model can help learn representative features for downstream tasks.