CoCoS: Enhancing Semi-supervised Learning on Graphs with Unlabeled Data via Contrastive Context Sharing

Siyue Xie, Da Sun Handason Tam, Wing Cheong Lau

The Chinese University of Hong Kong
{xs019, tds019, wclau}@ie.cuhk.edu.hk

Abstract

Graph Neural Networks (GNNs) have recently become a popular framework for semi-supervised learning on graph-structured data. However, typical GNN models heavily rely on labeled data in the learning process, while ignoring or paying little attention to the data that are unlabeled but available. To make full use of available data, we propose a generic framework, **Co**ntrastive **Co**ntext **S**haring (CoCoS), to enhance the learning capacity of GNNs for semi-supervised tasks. By sharing the contextual information among nodes estimated to be in the same class, different nodes can be correlated even if they are unlabeled and remote from each other in the graph. Models can therefore learn different combinations of contextual patterns, which improves the robustness of node representations. Additionally, motivated by recent advances in self-supervised learning, we augment the context sharing strategy by integrating with contrastive learning, which naturally correlates intra-class and inter-class data. Such operations utilize all available data for training and effectively improve a model's learning capacity. CoCoS can be easily extended to a wide range of GNN-based models with little computational overheads. Extensive experiments show that CoCoS considerably enhances typical GNN models, especially when labeled data are sparse in a graph, and achieves state-of-the-art or competitive results in real-world public datasets.

Introduction

As an effective tool for learning graph-related patterns, Graph Neural Networks (GNNs) have been widely applied to analyze graph-structured data in recent years. Representations learned through GNN can be used in diverse tasks across different domains, such as traffic predictions (Li et al. 2017; Yu, Yin, and Zhu 2017), recommendations (Ying et al. 2018; Li et al. 2020), etc., which advances the development of the corresponding applications.

In general, typical GNNs (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2016; Veličković et al. 2017) generate node representations by aggregating information from a target node's neighbors in each layer. By stacking multiple layers, GNNs can extend the receptive field to observe more nodes. Topological information of a target node's *context*, which refers to an instance's neighborhood information (including node/ edge attributes and the graph topology), will be captured. Task-related patterns can then be extracted from the context by supervising the learned representations of labeled nodes.

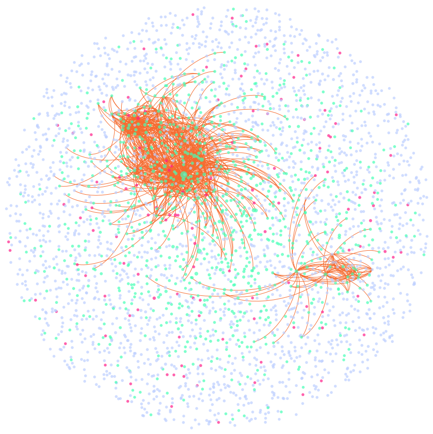

Figure 1: Visualization of the receptive field (in the Citeseer dataset) of a 2-layer GNN model. Red dots are labeled nodes for training, green dots are nodes in the receptive field of any labeled nodes, and blue dots are other unobserved nodes in the training stage. For visualization clarity, an edge is eliminated if one of its end point's degree is smaller than 10.

However, in many real-world cases, labelling is expensive and labeled nodes are usually sparse in the graph. Nodes located in the receptive field of labeled nodes only account for a small proportion of a graph. An example in the Citeseer dataset is shown in Fig. 1. Following the general scheme of GNNs, data are not well utilized since many unlabeled nodes hardly participate into the learning process. Although we can build a GNN model with deeper layers to cover more available nodes, the performance often degrades due to the over-smoothing problem (Rong et al. 2019). This indicates that simply stacking multiple GNN layers may not be effective to fully utilize those unlabeled data. On the other hand, ignoring the unlabeled data may fail to capture some distinct or discriminative patterns of the graph, which results in limited representation capacity of the model. Such observations inspire us to explore reasonable ways to learn from unlabeled data by correlating the unlabeled data with those labeled under the semi-supervised setting.

In recent years, contrastive learning has attracted much attention from researchers in different domains (Hjelm et al. 2018; He et al. 2020). Different from the semi-supervised setting, models trained through contrastive learning do not rely on any labeled data. By designing feasible augmentations, general patterns of data can be learned from contrastive pairs/ groups and generalized to different downstream tasks. Some recent works also introduce contrastive learning to analyze graph-structured data (Thakoor et al. 2021; Zhu et al. 2020). Since all available data can be incorporated for training, some approaches even outperform other semi-supervised counterparts in several tasks (Hassani and Khasahmadi 2020; Veličković et al. 2018).

To better leverage unlabeled data, we propose a generic framework **Co**ntrastive **Co**ntext **S**haring (CoCoS), which enhances the learning capacity of GNN models under semi-supervised settings. Different from previous semi-supervised methods, CoCoS shares the contextual information within the same class of nodes, where class labels of unlabeled nodes are estimated by a pretrained GNN model and can be progressively updated throughout the learning process. Such a sharing scheme correlates both labeled and unlabeled nodes (even if two nodes are far away from each other in the graph) and induces different combinations of contextual patterns that may not be observed in the original graph. This enables the model to learn robust node representations. In addition, motivated by recent advances in self-supervised learning, we adopt similar ideas from contrastive learning to further enhance the context sharing strategy. Specifically, the context sharing scheme can be viewed as a feasible augmentation of the original graph. Contrastive pairs can therefore be constructed among the original graph and its different augmentations, which unifies intra-class samples (nodes with the same label) and inter-class samples (nodes with different labels) during the learning process. With these operations, all available data are well exploited and utilized for training. Compared with previous works on graph contrastive learning, CoCoS-enhanced GNNs can be jointly optimized in an end-to-end manner with supervision. The learned node representations can therefore be more task-oriented, which effectively improves the discriminative power of a model. CoCoS can be readily applied to a wide range of GNN-based models with little computational overheads. Extensive experiments show that CoCoS consistently improves the performance of different GNNs and achieves state-of-the-art or competitive results on node classification tasks.

In general, the contributions of our proposed method can be summarized as follows:

1. We propose the context sharing scheme for GNN models under semi-supervised settings. By properly incorporating unlabeled data into the learning process, GNN models can learn more discriminative contextual patterns.
2. We build a generic framework, CoCoS, by integrating contrastive learning with the context sharing strategy. All available data across different classes can be well utilized during training, which improves the learning capacity of a model.
3. CoCoS can be readily adapted to different GNN-based models. Extensive experiments demonstrate its superiority over other GNN baselines and state-of-the-art graph contrastive learning frameworks. The code of CoCoS is available online [1].

Related Works

Semi-supervised Learning with GNNs

Recent works on graph learning employ GNNs to generate powerful node representations. A typical work is GC-N (Kipf and Welling 2016), which captures graph structural information by iterative aggregations of neighbors. GraphSAGE (Hamilton, Ying, and Leskovec 2017) propose a message-passing framework to adapt to large-scale graph-s, which is followed by many other works (Xu et al. 2018b; Wu et al. 2019; Xu et al. 2018a; Klicpera, Bojchevski, and Günnemann 2018; Klicpera, Weißenberger, and Günnemann 2019). GAT (Veličković et al. 2017) and GaAN (Zhang et al. 2018) introduce attention mechanism into aggregation to further highlight important neighbors. To improve data u-tilization, some works additionally apply self-training (Sun, Lin, and Zhu 2020; Stretcu et al. 2019) or post-processing (Huang et al. 2020) to enhance GNN models. Others (Feng et al. 2020; Wang et al. 2020) regularize GNN by consistency constraints. However, these approaches rely on prior human knowledge, or fail to utilize all information available in the graph when labeled nodes are sparse, which prevents GNNs from learning more powerful representations. More discussions refer to supplementary materials [2].

Contrastive Learning on Graphs

Given the limited amount of ground-truth labels available in real-world graph-structured data, unsupervised or self-supervised training of GNNs has become an active research direction, where contrastive learning is the currently dominant approach. In general, Graph Contrastive Learning (G-CL) approaches learn node representations by training a graph encoder and discriminator. The graph encoder produces node representations, while the discriminator distinguishes representation pairs that are semantically similar from those dissimilar. For instance, DGI (Veličković et al. 2018) contrasts node-local patches and graph-global representations. InfoGraph (Sun et al. 2019) extends DGI to the multi-graph and graph classification settings. GRACE (Zhu et al. 2020) and BGRL (Thakoor et al. 2021) use node features masking and random edges removal to generate augmented views of a graph. MVGRL (Hassani and Khasahmadi 2020) follows the idea of DGI on contrasting but uses graph diffusion to generate augmented graphs. Due to better utilization of unlabeled data, the performance of advanced GCL approaches are comparable to semi-supervised methods or even reach the state-of-the-art.

[1]http://github.com/XsLangley/CoCoS

[2]The supplementary materials can be found in the corresponding GitHub code repository.

Depth	Cora	Citeseer	Pubmed
1	76.22	86.71	98.20
2	38.55	67.18	85.81
3	18.09	50.32	53.03

Table 1: The ratio of unobserved data (nodes) regarding different depths (layers / hops) of GNNs (%).

Preliminaries and Motivations

Problem Formulation

A graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ is the set of nodes in the graph, and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set. Nodes in the graph are associated with features denoted by $\boldsymbol{X} = \{\boldsymbol{x}_v \in \mathbb{R}^{d_{in}} | \forall v \in \mathcal{V}\}$, where d_{in} is the dimension of the node feature vector. For a general GNN model, we aim to learn a mapping function $\boldsymbol{H} = GNN(\mathcal{G})$, where $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N]^\intercal \in \mathbb{R}^{N \times d_h}$ is the representation matrix. Each row of $\boldsymbol{H}$ is the learned embedding of a node. Under the semi-supervised setting, some nodes are annotated with labels while the remaining are unlabeled, i.e. $\mathcal{V} = \{\mathcal{V}^L, \mathcal{V}^U\}$, where $\mathcal{V}^L$ and $\mathcal{V}^U$ are the labeled and unlabeled nodes set respectively. The model can therefore be trained in a semi-supervised manner by the annotated nodes using labels $\boldsymbol{Y}^L = \{\boldsymbol{y}_v | v \in \mathcal{V}^L\}$ throughout the learning process, where $\boldsymbol{y}_v \in \mathbb{R}^K$ is a K-dimensional (class) one-hot vector for node v. Under the semi-supervised setting, our goal is to learn $GNN(\mathcal{G})$ to predict the labels of other unlabeled nodes correctly.

Preliminary Analyses on Data Utilizations

In many real-world scenarios, annotated data are precious resources as data annotation is a challenging and expensive task. Thus, labeled nodes for semi-supervised training are commonly sparse in the graph. For instance, Cora, Citeseer and Pubmed datasets (Sen et al. 2008; Namata et al. 2012), which are widely used for GNNs performance evaluations (Kipf and Welling 2016; Wu et al. 2019), have only 5.17%, 3.62%,and 0.30% labeled nodes for training respectively. Although the aggregation operations in GNN layers effectively extend the receptive field of the model, the proportion of observed nodes during model training is still limited.

We visualize the receptive field of a general 2-layer GNN model (e.g., GCN) in Fig. 1, where observed nodes (including labeled nodes and their neighbors within two hops) only account for 32.82% of all nodes in the graph. We also list the ratio of unobserved data of a general GNN concerning different depths in Table 1, and we can observe that most available data are not observed in the learning process, which means the model does not learn any knowledge from these unobserved nodes. Although extending the receptive field by stacking more layers can incorporate more unlabeled nodes in training, GNN models will suffer from over-smoothing (Rong et al. 2019) when going deep, which often degrades the performance. Empirical experiments of some previous works (Kipf and Welling 2016; Veličković et al. 2017; Wu et al. 2019) also demonstrate that their GNN models perform the best in the aforementioned three datasets when stacking

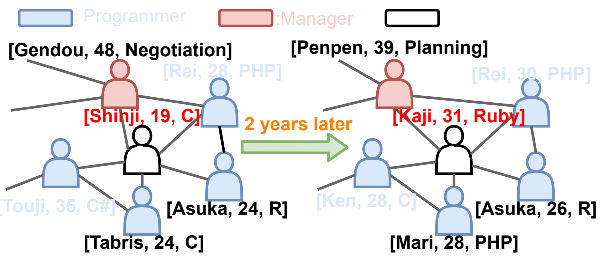

Figure 2: An example of consistent context for job position inference: the relation network in a company, where each staff is associated with a 3-dimensional vector indicating their names, ages and skills. The person in each position may change as time goes. Even though its context information fluctuates, we are confident to infer that the job position of the central node should be a programmer.

only two GNN layers. This inspires us to explore other ways to mine useful information from unlabeled data instead of merely stacking a deeper model.

Context Sharing Using Oracles

Based on the above analyses, it is clear that unlabeled nodes are usually under-exploited in semi-supervised scenarios. Therefore, it will be promising to improve GNN's learning capacity by mining useful information from those ignored nodes. Our idea is to correlate unlabeled nodes with the labeled ones, which is motivated by some heuristic observations.

Under the semi-supervised setting, we argue that the role (label) of a node can be inferred through its context topology (how the node connects with others) and the features/ attributes within its context (how all nodes look like). *Context* of a node v can be defined as the neighborhood information (including node/ edge attributes and the graph topology) around v. Moreover, in most real-world cases, the semantics of the context is kept consistent if some context information only slightly fluctuates within the same class of nodes. An illustrative example is shown in Fig. 2, where people join and leave a company from time to time, but job positions are usually fixed. Since GNN works by aggregating information from the neighborhoods, it can still infer the role of a node using such consistent contexts. Therefore, we may expect a GNN to learn more discriminative and robust features if contextual information can be shared within the same class of nodes (including those labeled and unlabeled).

We conduct some experiments to investigate the above presumption. Suppose we know the oracles, i.e., the ground-truth labels of both labeled and unlabeled nodes, of the graph in advance. The nodes set $\mathcal{V}$ can therefore be represented as $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_K\}$, where $\mathcal{V}_k$ $(k \in \{1, 2, ..., K\})$, is a node subset containing all nodes in the k-th class. We let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K$ be their corresponding node feature matrix respectively. We then randomly shuffle each nodes subset (as Step 2 of Fig. 3 shows), which in effect shuffles each node subset feature matrix, but keep the graph topology. This generates a graph with consistent contexts of the

Model	Cora	Citeseer	Pubmed
GCN	82.12	71.07	78.79
GCN-CS(O)	87.45	78.22	83.09

Table 2: Node classification performances with oracles (%).

original graph:

$$\tilde{\mathcal{V}}_k, \tilde{\boldsymbol{X}}_k = FeatShuf(\mathcal{V}_k, \boldsymbol{X}_k), \forall k, \tag{1}$$

where $FeatShuf(\cdot)$ is the random shuffle function on feature vectors. We define $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{\boldsymbol{X}})$ as the *context-consistent graph* generated by context sharing where $\tilde{\mathcal{V}} = \{\tilde{\mathcal{V}}_1, \tilde{\mathcal{V}}_2, \dots, \tilde{\mathcal{V}}_K\}$ is the vertex set of $\tilde{\mathcal{G}}$ and $\tilde{X}$ is the corresponding node feature matrix constructed from $\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_K$. (Note that the elements in $\tilde{\mathcal{V}}_k$ are the same as $\mathcal{V}_k$, but we use notation $\tilde{\mathcal{V}}_k$ to specifically indicate the nodes in $\tilde{\mathcal{G}}$.) Therefore, we can train a GNN model on the context-consistent graph to learn contextual patterns that are not available from the original graph. To align with common semi-supervised settings and make the result 'comparable', the loss function will only be computed based on the labeled nodes set $\mathcal{V}^L$ using cross-entropy:

$$\mathcal{L}_O = -\frac{1}{|\mathcal{V}^L|} \sum_{\tilde{v} \in \mathcal{V}^L} \boldsymbol{y}_{\tilde{v}}^T \log(\tilde{\boldsymbol{p}}_{\tilde{v}}), \;\; \tilde{\boldsymbol{p}}_{\tilde{v}} = Softmax(\tilde{\boldsymbol{h}}_{\tilde{v}}) \tag{2}$$

where $\tilde{\boldsymbol{h}}_{\tilde{v}}$ is the representation of node $\tilde{v} \in \tilde{\mathcal{V}}$ learned based on $\tilde{G}$. $\tilde{\boldsymbol{p}}_{\tilde{v}}$ is the predicted probability vector of node $\tilde{v}$ obtained by applying the Softmax function to $\tilde{\boldsymbol{h}}_{\tilde{v}}$.

Experimental settings follow the work of (Kipf and Welling 2016), and the details can be found in our supplementary materials. Note that the context-consistent graph is not unique so that we can generate a new $\tilde{\mathcal{G}}$ for the model in each training epoch. We take GCN (Kipf and Welling 2016) as an example to be the backbone model to evaluate the context sharing strategy, which is denoted as GCN-CS(O) ('O' stands for oracles). Comparisons of node classification accuracies are shown in Table 2. It is obvious that the classification accuracy is significantly improved by a large margin, which aligns with expectations. We also apply context sharing to other GNN backbones and it shows consistent improvement, where results and more analysis can be found in our supplementary materials. We may attribute the great effectiveness to the use of context-consistent graph, which explicitly correlates the labeled and unlabeled nodes within the same class. Since the context-consistent graph changes during training, GNN can learn to capture different reasonable combinations of context patterns. This encourages the model to learn robust representations for a node. In addition, the sharing operation is not restricted to the local neighborhoods. The context of two nodes remote from each other can be bridged by this strategy, which encourages the GNN to learn common knowledge of nodes within the same class.

Note that this is an unfair comparison as we use oracles of unlabeled nodes for intra-class feature random shuffling. However, it is still promising to further explore this direction since labels of unlabeled nodes are not directly used in the forward and backward propagation.

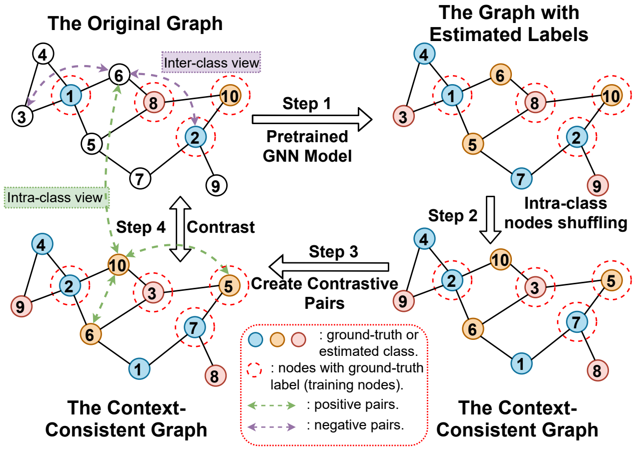
The Original Graph
Inter-class view
Intra-class view
Step 1
Pretrained GNN Model
The Graph with Estimated Labels
Step 2
Intra-class nodes shuffling
Step 3
Create Contrastive Pairs
Step 4
Contrast
The Context-Consistent Graph
The Context-Consistent Graph
: ground-truth or estimated class.
: nodes with ground-truth label (training nodes).
: positive pairs.
: negative pairs.

Figure 3: The framework of CoCoS. Nodes in a graph are uniquely indexed and partially labeled. A pretrained model is first used to estimate the label of each node. Context sharing is then applied to generate a context-consistent graph using the estimated labels. After that, contrastive pairs are constructed between the original and the context-consistent graph, which are used to enhance the learning capacity of the pretrained model.

Proposed Methods

With the above observations, we propose **Co**ntrastive **Co**ntext **S**haring (CoCoS) to enhance the learning of GNNs under the semi-supervised setting. An overview of CoCoS is shown in Fig. 3. In this section, we specify the details of this framework.

Labels Estimation using Pretrained Models

Preliminary experiments in the previous section have shown the potential benefits of context sharing for training GNNs. However, in preliminary experiments, ground-truth labels of unlabeled nodes are used for intra-class shuffling, while the oracle information is actually inaccessible in practical settings. In CoCoS, we circumvent such a difficulty by using a pretrained GNN model. Specifically, given a partially annotated graph, a GNN can be trained using these limited labeled nodes. Such a pretrained GNN model can give us an initial estimation of class labels of every node in the graph, as shown in 'Step 1' of Fig. 3. For nodes in the labeled set $\mathcal{V}^L$ (marked by the red circle with the dotted line in Fig. 3), we keep their ground-truth labels instead of their estimations. Although such an estimation can be noisy, it is demonstrated to be effective in our following empirical experiments. In addition, the estimated labels will be progressively overridden during training, which corrects some wrong predictions by the initial pretrained model and reduces the impact of noises.

Intra-class Context Sharing with Estimated Labels

With the ground-truth and estimated labels for each node, we can then apply context sharing to the original graph. As 'Step 2' shown in Fig. 3, nodes estimated to have the same class labels will be randomly shuffled. We apply intra-class feature shuffling (Eq. 1) to generate a new context-consistent graph. Since the shuffling operation involves all available data, different unlabeled nodes can be exchanged with labeled nodes or nodes in the neighborhoods of labeled nodes. This helps the model make full use of all available data and therefore learn more diverse and general context-aware patterns for the downstream task.

Learning Enhancement with Contrastive Pairs

By context sharing, we are able to enhance the learning capacity by fine-tuning the pretrained GNN model on the context-consistent graph following Eq. 2. In addition to that, we further enhance the model by contrasting the context-consistent graph and the original graph, which is motivated by some recent advances in graph contrastive learning (GCL) (Veličković et al. 2018; Hassani and Khasahmadi 2020).

Specifically, we draw an analogy between context sharing and augmentations in contrastive learning. Since the context-consistent graph keeps similar semantics, it can be regarded as an augmentation view of the original graph. Instead of contrasting between the graph-level and the node-level representations, which is common in many previous GCL methods (Veličković et al. 2018; Hassani and Khasahmadi 2020), CoCoS contrasts between two node representations. Following the notations in the last section, we denote $\boldsymbol{H} = \{\boldsymbol{h}_{v_1}, \boldsymbol{h}_{v_2}, ..., \boldsymbol{h}_{v_N}\} = GNN(\mathcal{G})$ and $\tilde{\boldsymbol{H}} = \{\tilde{\boldsymbol{h}}_{\tilde{v}_1}, \tilde{\boldsymbol{h}}_{\tilde{v}_1}, ..., \tilde{\boldsymbol{h}}_{\tilde{v}_N}\} = GNN(\tilde{\mathcal{G}})$ as the node representations matrix learned from the original and context-consistent graph respectively, where each entry corresponds to a node position. Since context is consistent in the same node position, $\tilde{\boldsymbol{h}}_{\tilde{v}_i}$ can be regarded as an augmentation of $\boldsymbol{h}_{v_i}$. Therefore, $(\boldsymbol{h}_{v_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_i})$ can be regarded as a positive pair between the original and the context-consistent graph. In addition, for nodes with the same estimated class labels, their representations can also be regarded as positive pairs. Thus, we can build four kinds of intra-class positive pairs for contrasting:

$$(\boldsymbol{h}_{v_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_i}); \tag{3}$$

$$(\boldsymbol{h}_{v_i}, \boldsymbol{h}_{v_j}) : \boldsymbol{y}_{v_i} = \boldsymbol{y}_{v_j}, i \neq j; \tag{4}$$

$$(\tilde{\boldsymbol{h}}_{\tilde{v}_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_j}) : \boldsymbol{y}_{\tilde{v}_i} = \boldsymbol{y}_{\tilde{v}_j}, i \neq j; \tag{5}$$

$$(\boldsymbol{h}_{v_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_j}) : \boldsymbol{y}_{v_i} = \boldsymbol{y}_{\tilde{v}_j}, i \neq j; \tag{6}$$

where $\boldsymbol{y}_{v_i}$ (or $\boldsymbol{y}_{\tilde{v}_i}$) is the ground-truth label (or estimated label) of node v_i (or node $\tilde{v}_i$). Two nodes from different classes (inter-class) in the original graph are paired as a negative sample:

$$(\boldsymbol{h}_{v_i}, \boldsymbol{h}_{v_j}) : \boldsymbol{y}_{v_i} \neq \boldsymbol{y}_{v_j}, i \neq j; \tag{7}$$

We apply a Multi-Layer Perceptron (MLP) as the discriminator D to maximize the mutual information between the positive pairs:

$$D(\boldsymbol{z}_1, \boldsymbol{z}_2) = MLP(\boldsymbol{z}_1 || \boldsymbol{z}_2), \tag{8}$$

where $(\boldsymbol{z}_1, \boldsymbol{z}_2)$ is a pair of inputs, '$||$' is the concatenation operator and p is a scalar value indicating the agreement score. In CoCoS, we can use different combinations of different kinds of positive pairs in the training stage. As we will show in our ablation experiments, their performances vary in different datasets. We empirically pick Eq. (3) and (5) as the positive pairs. In each training epoch, we generate a context-consistent graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$ by intra-class feature shuffling (Eq. 1). Following the work of DGI (Veličković et al. 2018), we formulate the contrastive loss for each epoch by unifying both intra-class and inter-class pairs as follows:

$$\begin{aligned}\mathcal{L}_{ctr} = -\frac{1}{|\mathcal{V}^U|}\Bigg(&\frac{1}{2}\sum_{v_i \in \mathcal{V}^U} \log D(\boldsymbol{h}_{v_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_i}) \\ &+ \frac{1}{2}\sum_{\tilde{v}_i \in \mathcal{V}^U} \mathbb{E}_{\tilde{v}_j}\left[\log D(\tilde{\boldsymbol{h}}_{\tilde{v}_i}, \tilde{\boldsymbol{h}}_{\tilde{v}_j}) \mid \boldsymbol{y}_{\tilde{v}_j} = \boldsymbol{y}_{\tilde{v}_i}\right] \\ &+ \sum_{v_i \in \mathcal{V}^U} \mathbb{E}_{v_j}\left[\log\Big(1 - D(\boldsymbol{h}_{v_i}, \boldsymbol{h}_{v_j})\Big) \mid \boldsymbol{y}_{v_j} \neq \boldsymbol{y}_{v_i}\right]\Bigg)\end{aligned} \tag{9}$$

By applying contrastive learning, models are encouraged to learn discriminative representations to distinguish intra-class nodes from inter-class nodes.

Model Training

The final objective of CoCoS for training consists of two parts: a classification loss following Eq. 2 and the contrastive loss (Eq. 9). To better utilize the labeled data, we reformulate the classification loss of Eq. 2 as follows:

$$\mathcal{L}_{cls} = -\frac{1}{2|\mathcal{V}^L|}\left(\sum_{v \in \mathcal{V}^L} \boldsymbol{y}_v^T \log(\boldsymbol{p}_v) + \sum_{\tilde{v} \in \mathcal{V}^L} \boldsymbol{y}_{\tilde{v}}^T \log(\tilde{\boldsymbol{p}}_{\tilde{v}})\right) \tag{10}$$

Therefore, the objective for CoCoS can be formulated as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{ctr}, \tag{11}$$

where α is an adjustable hyperparameter.

Since CoCoS is a framework to enhance existing GNN models, we apply CoCoS to fine-tune the pretrained GNN that is used for labels estimation. For each training epoch, a new context-consistent graph will be generated for training. Estimated labels can be overridden by the prediction results of the current training model for every $\eta \in \mathbb{N}$ epochs.

Experiments and Analysis

Experimental Setup

We evaluate CoCoS by semi-supervised node classification tasks. Performances are compared with different GNN models and state-of-the-art GCL methods.

Datasets Our experiments are conducted on four real-world public datasets, including Cora, Citeseer, Pubmed and Ogbn-arxiv (Hu et al. 2020). All these four datasets are citation networks. For Cora, Citeseer and Pubmed, we use the same train, validation, test split according to (Kipf and Welling 2016). For Ogbn-arxiv, we use the official split for training and evaluations. More details on dataset descriptions can be found in our supplementary materials.

Model	Cora	Citeseer	Pubmed	Ogbn-arxiv
MLP	58.51	55.64	72.71	$55.50 \pm 0.23^{\dagger}$
n2v	72.35	50.82	62.03	-
n2v+feat	-	-	-	$70.07 \pm 0.13^{\dagger}$
DGI*	$82.30^{\dagger}$	$71.80^{\dagger}$	$76.80^{\dagger}$	-
BGRL*	-	-	-	$71.57 \pm 0.11^{\dagger}$
MVGRL*	$\mathbf{86.80}^{\dagger}$	$73.30^{\dagger}$	$80.10^{\dagger}$	-
GCN	82.12	71.07	78.79	71.51 ± 0.14
GAT	81.95	70.68	78.29	72.24 ± 0.16
SAGE(g)	82.06	70.66	78.65	71.41 ± 0.12
SAGE(m)	81.55	69.76	78.49	71.49 ± 0.09
JK-Net	80.69	68.03	78.65	71.49 ± 0.15
SGC	80.55	72.00	78.86	69.99 ± 0.18
CoCoS	84.15	**73.57**	**80.92**	$\mathbf{72.95} \pm 0.19$

Table 3: Accuracies of node classifications (%). Values marked by '†' are from the original papers or the OGB official leaderboard. Models marked by '$*$' have different network settings with our implementations.

Models for Comparison In our experiments, we include four types of methods for comparison: a model learned without graph information - a multiplayer perceptron (MLP), unsupervised graph learning models without GNNs (node2vec (Grover and Leskovec 2016), node2vec+node features), GCL methods (DGI (Veličković et al. 2018), BGRL (Thakoor et al. 2021), MVGRL (Hassani and Khasahmadi 2020)) and typical semi-supervised GNNs (GCN (Kipf and Welling 2016), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Veličković et al. 2017), JK-Net (Xu et al. 2018b), SGC (Wu et al. 2019)).For GraphSAGE, we implement its two variants with different aggregators (gcn and mean aggregator), denoted as SAGE(g) and SAGE(m) respectively. We reimplement all the compared models for experiments except the group of GCL methods. For GCL methods, we directly compare with the results reported in their paper. More details on model descriptions can be found in the supplementary materials.

Implementations We use GCN as the pretrained model for Cora, Citeseer and Pubmed, where the network architecture follows the original paper (Kipf and Welling 2016), i.e., a two-layer model with 16-dimensional hidden layers. Since the variants of GAT (Veličković et al. 2017) perform significantly better in the OGB leaderboard, we take GAT as the backbone for Ogbn-arxiv. We follow the grid-search strategy to find the optimal hyperparameters, and we set $\alpha = 0.6$ in Eq. 11 in training. For Cora, Citeseer and Pubmed, we override the estimated labels for every ten training epochs ($\eta = 10$), while $\eta = 4$ in Ogbn-arxiv. More details about implementations refer to our supplementary materials.

Experimental Results on Node Classifications

The experimental results are shown in Table 3. We also report additional validation accuracies of Ogbn-arxiv in the supplementary materials.

From the results, we can clearly observe that CoCoS outperforms all other competitors in Citeseer, Pubmed, and Ogbn-arxiv datasets and be the runner-up in Cora. Note that the best model in Cora, i.e., MVGRL, applies a wider GNN backbone (with 512-dim hidden layers) for training (while CoCoS is only configured with 16-dim hidden layers to align with other semi-supervised baselines). Therefore, MVGRL uses a much larger model to learn from more complex data. Similar settings are applied to other GCL methods. Even so, CoCoS still yields competitive results, which demonstrates its effectiveness. We also conduct an experiment following MVGRL's network settings, where results and more comparisons are reported in the supplementary materials.

Compared with other semi-supervised methods, the superiority of CoCoS is more conspicuous. The classification accuracies are improved by over 2% in Cora, Citeseer and Pubmed (vs. GCN), and nearly 1% in the Ogbn-arxiv (vs. GAT). We attribute this improvement to the better utilization of unlabeled data. Since all available data can be incorporated for training, CoCoS is capable to capture more patterns, which yields a powerful embedding for each node. Note that the improvements in the first three datasets are more significant than that in Ogbn-arxiv. This may be due to the fact that about 53.70% of nodes are labeled for training in Ogbn-arxiv (while this value is no more than 5.2% in the other three datasets). By stacking a multi-layer GNN model, the receptive field can nearly cover the whole graph. Even so, CoCoS can still improve the performance of the pretrained model (GAT), which verifies that the benefit is not only brought by training on more available nodes, but also by learning discriminative information from data. We also evaluate CoCoS on four additional datasets, where results and analyses refer to the supplementary materials.

Model	Cora	Citeseer	Pubmed
CoCoS-c	**84.15**(+2.03)	**73.57**(+2.50)	80.92(+2.13)
CoCoS-a	83.81(+1.86)	72.98(+2.30)	79.85(+1.56)
CoCoS-sg	83.77(+1.71)	73.10(+2.44)	**81.59**(+2.94)
CoCoS-sm	83.58(+2.03)	73.44(+3.68)	80.68(+2.19)
CoCoS-j	83.59(+2.90)	71.47(+3.44)	80.68(+2.03)
CoCoS-sc	81.33(+0.78)	72.39(+0.39)	79.35(+0.49)

Table 4: The classification accuracies of CoCoS by using different GNN backbones (%). Values in the parentheses are improvements with respect to the corresponding baselines.

Ablation Studies

Apart from evaluating the proposed framework, we also investigate each module/ operation that is adopted in CoCoS.

Enhancement for Different GNNs In this experiment, we evaluate the generalization ability of CoCoS by applying it to different GNN models. Concretely, different variants of CoCoS are denoted by CoCoS-c (GCN-based), CoCoS-a (GAT-based), CoCoS-sg (SAGE(g)-based), CoCoS-sm (SAGE(m)-based), CoCoS-j (JK-Net-based) and CoCoS-sc (SGC-based).

The experimental results are shown in Table 4. Obviously, all GNN models (except SGC) obtain different levels of performance gain by integrating with CoCoS. Improvements are significant for SAGE(g) in Pubmed and SAGE(m) as well as JK-Net in Citeseer, where CoCoS-sg performs even

Model	Cora	Citeseer	Pubmed	Ogbn-arxiv
GCN	23.1K	59.4K	8.1K	109.6K
CoCoS-c	24.1K	60.3K	8.6K	130.6K
GAT	92.4K	237.6K	32.4K	144.2K
CoCoS-a	93.4K	238.5K	32.9K	146.3K
SAGE(g)	23.1K	59.4K	8.1K	109.6K
CoCoS-sg	24.1K	60.3K	8.6K	130.6K
SAGE(m)	46.1K	118.7K	16.1K	218.2K
CoCoS-sm	47.1K	119.6K	16.6K	239.1K
JK-Net	23.4K	59.7K	8.4K	196.4K
CoCoS-j	24.5K	60.6K	8.9K	217.4K
SGC	10.0K	22.2K	1.5K	5.2K
CoCoS-sc	11.1K	23.1K	2.0K	26.2K

Table 5: The size of learnable parameters of each model.

Model	Cora	Citeseer	Pubmed
GCN-CS	82.83(+0.71)	72.30(+1.23)	78.93(+0.14)
GCN-Both	83.20(+1.08)	73.13(+2.06)	79.68(+0.89)
CoCoS	84.15(+2.03)	73.57(+2.50)	80.92(+2.13)

Table 6: The classification accuracies when using different context sharing strategies (%). Values in the parentheses are the improvements with respect to GCN.

better than CoCoS-c in the Pubmed. Such results demonstrate that CoCoS is universally valid for different GNNs.

We also compare the computation overheads of CoCoS by the model size. Table 5 lists the size of learnable parameters of models. Compared with the original baseline, the increment of model size is marginal. This fact shows that CoCoS is not only effective in performance improvement but also friendly to model extension. Note that the model size of SGC is much smaller than other counterparts as trainable parameters are only available at the last output layer, which limits its representation abilities on learning. This may explain why SGC only improves slightly (or even get worse in Ogbn-arxiv) using CoCoS.

Performance of Context Sharing

In this experiment, we assess the effect of context sharing. Specifically, we remove the contrastive loss, i.e., Eq. 9, from Eq. 11. The model is then trained respectively by two different kinds of classification loss, i.e., Eq. 2 and Eq. 10, to evaluate the performance. We use GCN as our backbone (pretrained) model. The model trained using Eq. 2 is denoted as GCN-CS and the one trained by Eq. 10 is denoted as GCN-Both. Experiments are conducted on Cora, Citeseer and Pubmed datasets.

Experimental results based on GCN are shown in table 6. More results of context sharing using other GNN models are included in the supplementary materials. Obviously, the baseline model, GCN, can be consistently improved by using different strategies of context sharing. Moreover, GCN-Both always performs better than that of GCN-CS. We owe it to the introduction of the fully-supervised part (first term) in Eq. 10. Since the context sharing is operated based on the estimated labels, which can be different from oracles, it inevitably introduces noises to the learning process. In addition, the shuffling operation brings randomness to the model, which may make it unstable during training. In contrast, the fully-supervised part forces the model to learn from the original graph with fixed inputs. Such an operation helps stabilize the training and correct some wrong estimations, which balances between generalization and specialization.

Model	Cora	Citeseer	Pubmed
CoCoS-F	**84.49**(+2.37)	**73.62**(+2.55)	80.93(+2.14)
CoCoS-T	84.07(+1.95)	73.39(+2.32)	**81.24**(+2.45)
CoCoS-S	84.18(+2.06)	73.38(+2.31)	81.03(+2.24)
CoCoS-M	84.48(+2.36)	73.57(+2.50)	80.39(+1.60)
CoCoS-FS	84.15(+2.03)	73.57(+2.50)	80.92(+2.13)
CoCoS-MS	84.02(+1.90)	73.56(+2.49)	80.81(+2.02)

Table 7: The classification accuracies when training with different contrastive pairs (%). Values in the parentheses are improvements with respect to GCN.

Performance of Different Contrastive Augmentations In Table 6, by comparing CoCoS with GCN-CS and GCN-Both, we find that it is beneficial to introduce contrastive pairs into training. As shown in Eq. 3 to 6, we have four candidates for contrastive pairs. To further investigate the effect of different contrastive augmentations, we enhance GCN by different (combinations of) positive contrastive pairs. Specifically, we use 'F', 'T', 'S', and 'M' to mark the contrastive pairs from Eq. 3 to 6 respectively. For example, CoCoS-FS stands for the model trained using the combination of Eq. 3 and 5.

Table 7 shows the experimental results. We can observe that all different candidates of positive contrastive pairs are effective for learning enhancement, where accuracies vary slightly across different variants, which demonstrates the robustness of CoCoS. We also conduct similar experiments on other GNN baselines. These results are reported in the supplementary materials. It is worth mentioning that GraphSAGE-based variants usually perform better than other counterparts in the Pubmed dataset, with the highest accuracy of 82.06%. This reveals that CoCoS can be more effective to some models in specific datasets.

Conclusion

In this paper, we argue that available data in graph semi-supervised learning are usually under-exploited, which restricts the learning capacity of GNNs. To handle this problem, we propose CoCoS, a generic framework to enhance existing GNN-based models. Motivated by the recent advances in contrastive learning, CoCoS applies context sharing to the given graph and builds contrastive pairs for training, encouraging GNN models to better manage available data and learn discriminative representations for downstream tasks. Results of extensive experiments and ablation studies validate the effectiveness of CoCoS. Our future works will focus on differentiating the confidence/ importance of each label estimation, which is expected to reduce noises for context sharing. Extending CoCoS to fully unsupervised settings will also be promising.

Acknowledgements

This research is supported in part by the Innovation and Technology Committee of HKSAR under the project#ITS/244/16 and the CUHK MobiTeC R&D Fund. We would like to thank Yikang Chen for the precious suggestions on this work.

References

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph Random Neural Network for Semi-Supervised Learning on Graphs. In *NeurIPS'20*.

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

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, 4116–4126. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*.

Huang, Q.; He, H.; Singh, A.; Lim, S.-N.; and Benson, A. R. 2020. Combining label propagation and simple models out-performs graph neural networks. *arXiv preprint arXiv:2010.13993*.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*.

Klicpera, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion improves graph learning. *Advances in Neural Information Processing Systems*, 32: 13354–13366.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Li, Z.; Shen, X.; Jiao, Y.; Pan, X.; Zou, P.; Meng, X.; Yao, C.; and Bu, J. 2020. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 1677–1688. IEEE.

Namata, G.; London, B.; Getoor, L.; Huang, B.; and EDU, U. 2012. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, 1.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2019. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Stretcu, O.; Viswanathan, K.; Movshovitz-Attias, D.; Platanios, E.; Ravi, S.; and Tomkins, A. 2019. Graph Agreement Models for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 32.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2019. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Sun, K.; Lin, Z.; and Zhu, Z. 2020. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5892–5899.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped representation learning on graphs. *arXiv preprint arXiv:2102.06514*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; Liu, J.; and Hooi, B. 2020. Nodeaug: Semi-supervised node classification with data augmentation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 207–217.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018a. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018b. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, 5453–5462. PMLR.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 974–983.

Yu, B.; Yin, H.; and Zhu, Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Zhang, J.; Shi, X.; Xie, J.; Ma, H.; King, I.; and Yeung, D.-Y. 2018. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.