TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 23/24 pp605–616
DOI: 10.26599/TST.2023.9010071
Volume 29, Number 2, April 2024

Few-Shot Graph Classification with Structural-Enhanced Contrastive Learning for Graph Data Copyright Protection

Kainan Zhang, DongMyung Shin, Daehee Seo, and Zhipeng Cai*

Abstract: Open-source licenses can promote the development of machine learning by allowing others to access, modify, and redistribute the training dataset. However, not all open-source licenses may be appropriate for data sharing, as some may not provide adequate protections for sensitive or personal information such as social network data. Additionally, some data may be subject to legal or regulatory restrictions that limit its sharing, regardless of the licensing model used. Hence, obtaining large amounts of labeled data can be difficult, time-consuming, or expensive in many real-world scenarios. Few-shot graph classification, as one application of meta-learning in supervised graph learning, aims to classify unseen graph types by only using a small amount of labeled data. However, the current graph neural network methods lack full usage of graph structures on molecular graphs and social network datasets. Since structural features are known to correlate with molecular properties in chemistry, structure information tends to be ignored with sufficient property information provided. Nevertheless, the common binary classification task of chemical compounds is unsuitable in the few-shot setting requiring novel labels. Hence, this paper focuses on the graph classification tasks of a social network, whose complex topology has an uncertain relationship with its nodes' attributes. With two multi-class graph datasets with large node-attribute dimensions constructed to facilitate the research, we propose a novel learning framework that integrates both meta-learning and contrastive learning to enhance the utilization of graph topological information. Extensive experiments demonstrate the competitive performance of our framework respective to other state-of-the-art methods.

Key words: few-shot learning; contrastive learning; data copyright protection

1 Introduction

While big data are driving machine learning to new heights, the performance of machine learning is often limited by the availability of input samples. Open-source licenses can promote the sharing and use of data, allowing more data to be available for machine learning training, but sharing data under open-source licenses is not always appropriate, particularly when handling sensitive or personal information like social network data[1–3]. Certain open-source licenses may not provide adequate protection for such data, while other data may be legally or regulatory restricted, making sharing impossible regardless of the licensing model used[4–6]. Consequently, acquiring large amounts of labeled data can be a daunting, time-consuming, or costly task in many real-world situations[7]. Thus, researchers have developed the concept of Few-Shot

• Kainan Zhang and Zhipeng Cai are with Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA. E-mail: kzhang16@student.gsu.edu; zcai@gsu.edu.
• DongMyung Shin is with LSWare Inc., Seoul 08504, Republic of Korea. E-mail: dmshin24@naver.com.
• Daehee Seo is with College of Intelligence Information Engineering, Sangmyung University, Seoul 03016, Republic of Korea. E-mail: daehseo@smu.ac.kr.
* To whom correspondence should be addressed.
Manuscript received: 2023-06-09; revised: 2023-07-12; accepted: 2023-07-15

Learning (FSL) inspired by the rapid reasoning ability of human beings. FSL involves learning from a small number of samples and is well-suited for situations where large amounts of labeled data may be unavailable. Unlike standard supervised learning, where plenty of training data are available, FSL must distinguish, instead of recognizing, the new samples by learning a similarity metric and generalizing the model for broad applications. This is particularly advantageous when dealing with copyrighted data that may have restricted sharing or distribution rights. Since few-shot learning requires only a small number of labeled examples, it reduces the need to share the entire dataset, thus minimizing the risk of copyright infringement.

The success of FSL in image processing has paved the way for its application in diverse fields, including graph analytics. Although humans can quickly distinguish between images, distinguishing between complex graphs is still challenging, which makes graph learning crucial. Graph Neural Networks (GNNs), a powerful machine learning tool, utilize graph structure and node content information as inputs to perform various tasks related to graph analytics, such as edge-level, node-level, or graph-level tasks[8, 9]. The graph classification problem, which relates to GNNs with graph-level outputs, often employs pooling and readout operations. The pooling layer[10, 11] coarsens a graph into a subgraph, with node representations on the coarsened graph representing a higher graph-level representation. Subsequently, the readout layer aggregates the hidden representations of the subgraphs, computing a compact graph representation that serves as the class label(s) for the entire graph.

In a recent study[12], researchers compared various GNN methods for graph classification in a standardized and uniform evaluation framework. They pointed out that structure information has not been fully utilized in chemical and social datasets, and offered two possible reasons: either an effective solution exists without using topological information, or GNNs are not exploiting graph structure adequately. Since structural features are known to correlate with molecular properties in chemistry, a common binary classification task of chemical compounds (active or inactive) can be solved without GNNs if provided with sufficient molecular information (i.e., large node-attribute dimensions). However, this approach is unsuitable for FSL due to the few new classes. Therefore, we focus on FSL for social graphs, whose complex topology has an uncertain relationship with its nodes' attributes[13]. When performing multi-class classification tasks, the graph embedding generated by the model often needs to cover more comprehensive node information including its attributes and adjacency relationships with other nodes (i.e., graph structure). When the graph label is more closely related to the graph structure, some part of the high-dimensional node attributes that is inconsistent with the graph structure will affect the result of graph classification. However, we usually input the complete node attributes, as prepossessing and filtering high-dimensional node attributes are not suitable. Figure 1 illustrates that node attributes may harm graph classification accuracy if graph structure is not adequately considered. In Fig. 1, we use graphs to represent two customers' consumption records over a period (the link indicates that the products are purchased together). If we only consider the product features (i.e., node attributes), we may find that the two users have similar consumption habits for footwear. While through learning graph topology, we can acquire differentiable graph embeddings as more comprehensive and representative user profiles with high-similarity node attributes retained. The left customer shows a diversified consumption habit, while the right customer is more like a sneakerhead. The practical significance of this problem can be seen, for instance, in the cold-start issue caused by the small sample size limit in a recommendation

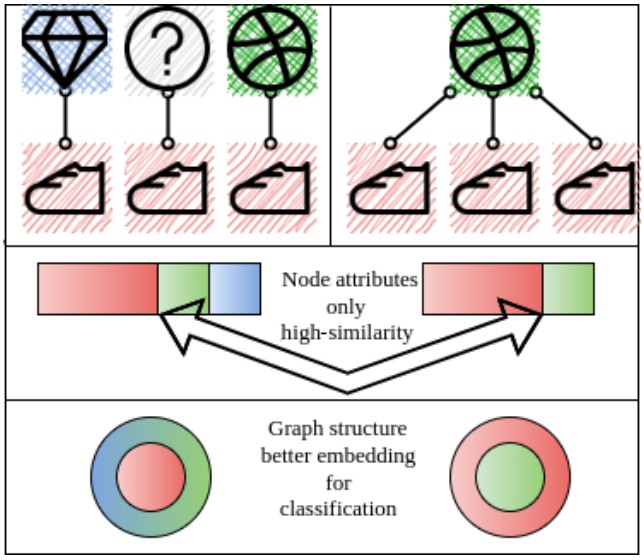


Fig. 1 A simple example that shows node attributes may hurt classification accuracy without adequately considering the graph structure. Strengthening graph structure learning can generate more distinguishable graph embedding while retaining high-similarity node information.

system[14].

Introducing a novel learning framework, Structural-Enhanced Graph Contrastive Learninig (SE-GCL), we aim to enhance the capabilities of GNNs in leveraging graph structure and addressing the challenges mentioned above. By utilizing meta-learning and contrastive learning techniques, our proposed framework achieves accurate graph classification results in an end-to-end process. The contributions of our work can be summarized as follows:

• We investigate a general scenario for few-shot graph classification tasks, and present a learning framework that integrates meta-learning and contrastive learning techniques. This integration allows us to achieve accurate graph classification results while ensuring the protection of data copyright.

• We construct two benchmark graph datasets with large node-attribute dimensions, designed for multi-class classification tasks. These datasets serve as valuable resources for future research in the field of few-shot graph classification. They enable researchers to evaluate and compare various algorithms and techniques within a context that respects data copyright protection.

• Through extensive experiments on ground truth datasets, we demonstrate the effectiveness of using contrastive learning techniques to enhance the utilization of graph topological information. Our framework achieves competitive performance when compared to other state-of-the-art methods.

2 Related Work

2.1 Few-shot learning and meta-learning

Wang et al.[15] defined FSL as a type of machine learning problem (specified by $\mathcal{E}$, $\mathcal{T}$, and $\mathcal{P}$), where the machine learns from experience $\mathcal{E}$ having limited supervised information to solve task $\mathcal{T}$ by improving the performance measure $\mathcal{P}$. Recently, meta-learning, also known as learning-to-learn, has become the most popular framework for FSL[16], as it has the significant benefit of discovering the consistency between the training objective and the test objective, allowing the model to learn from few-shot classification tasks directly[17]. Two meta-learners, matching network[18] and prototypical network[19], generate a memory component using the neural network. The former learns common representations for the labeled examples and matches the new test instance to the memorized examples via cosine similarity. The latter learns a prototype vector space for each class and assigns the test instance to the prototype by calculating the soft-max likelihood of the distance metric.

2.2 Few-shot learning on graph classification

Model-Agnostic Meta-Learning (MAML)[20] gains knowledge of a specific model parameter initialization that generates strong generalization performance on a new task with a limited number of gradient steps and a modest amount of training data. Ma et al.[21] extended this approach to a graph meta-learner, which uses GNN-based modules for fast adaptation on graph data and a step controller for the robustness and generalization of the meta-learner. Based on the spectrum of the graph's normalized Laplacian, Chauhan et al.[22] proposed few-shot graph classification using the latent inter-class relationships made by the super-graph, where the L_p Wasserstein distance serves as the metric for clustering the super graphs from prototype graphs. SMF-GIN[23] is another metric-based meta-learning framework for few-shot graph classification based on the graph isomorphism network[24], which explicitly considers the global structure and local structure of the input graph by the attention mechanism. Lately, Hassani[25] proposed an attention-based graph encoder that uses three congruent views of graphs to learn representations of task-specific information for fast adaptation as well as task-agnostic information for knowledge transfer.

2.3 Data copyright protection in graph contrastive learning

As its name suggests, Graph Contrastive Learning (GCL) contrasts graph samples and forces those with the same distribution toward one another in embedded space. Conversely, samples from different distributions are pressed against each other. Because contrastive learning[26] is the backbone of GCL, recent researches focus on exploring the design of graph augmentation schemes. Meanwhile, GCL must be employed within a framework that respects data ownership and copyright protection to ensure that the usage of graph data aligns with legal and ethical considerations. GCL[27] studies four graph augmentations to incorporate various priors for learning unsupervised representations of graph data. Based on their previous work, the authors proposed joint augmentation optimization[28] later, a unified bi-level optimization framework for automatically performing data augmentations. Similarly,

GCA[29] is another adaptive augmentation method that incorporates various priors for topological and semantic aspects of the graph. To generate more meaningful graph views, GRACE[30] develops a hybrid scheme on both structure and attribute levels to provide diverse node contexts, InfoGCL[31] follows the Information Bottleneck principle to reduce the mutual information between contrastive parts, and AD-GCL[32] avoids capturing redundant information in the graph views by optimizing adversarial graph augmentation strategies. The sampling bias is another common issue in GCL. Lin et al.[33] proposed a prototype-based clustering approach to mitigate the issue, while Yu et al.[34] discarded the graph augmentations and instead added uniform noises to enhance the uniformity of learned representations. With the increasing research on GCL, PyGCL[35] is developed as a bench-marking library to provide empirical evidence of the practical GCL algorithms and future research. It is essential to emphasize that advanced graph augmentation strategies can play a role in protecting data copyright by reducing the reliance on original data and generating synthetic data that reduce the reliance on a limited dataset, including copyrighted content, and enable the creation of larger, more diverse datasets. By having access to a broader range of data, organizations can reduce the need to use specific copyrighted content and minimize the risk of copyright infringement.

3 Proposed Framework

3.1 Problem definition

FSL is usually applied in supervised learning for object classification, also considered as N-way-K-shot classification. During the few-shot training phase, N categories (ways) with K samples (shots) per category are constructed as the support set. Then another batch of samples in N categories, named query set, is selected from the remaining data as the model's prediction object. The task is to distinguish these query set samples from the $N \times K$ support sets.

We formulate our few-shot graph classification problem as a standard N-way-K-shot classification task with appropriate data copyright protection measures, where a set of graphs $\{G_1, G_2, \ldots, G_m\}$ and their labels $\{y_1, y_2, \ldots, y_m\}$ are given. Let $G = (\boldsymbol{U}, \boldsymbol{E}, \boldsymbol{A}, y)$ denote an undirected unweighted graph, where $\boldsymbol{U}$ is the set of nodes, $\boldsymbol{E}$ is the set of edges, $\boldsymbol{A}$ is the set of node-attributes, and y is the label associated with the graph. According to the labels, $\{G_1, G_2, \ldots, G_m\}$ is split into $\{(G_{\text{train}}, y_{\text{train}})\}$ and $\{(G_{\text{test}}, y_{\text{test}})\}$ as the training set and test set, respectively. Notice that y_{train} and y_{test} must have no common classes for the meta-learning setting. In the meta-training phase, we construct the support dataset $\boldsymbol{D}_S(G_{\text{train}}, y_{\text{train}})$ by randomly selecting K samples from each of the N classes and the query dataset $\boldsymbol{D}_Q(G_{\text{train}}, y_{\text{train}})$ containing other M samples from the same N classes. The goal is to predict the label of each graph in the query dataset by giving a limited number of support graphs (i.e., $N \ll M$), which restricts access to the graph data to authorized individuals or entities involved and helps prevent unauthorized distribution or misuse of copyrighted data. At the meta-testing stage, the same classification task is performed on $\boldsymbol{D}_S(G_{\text{test}}, y_{\text{test}})$ and $\boldsymbol{D}_Q(G_{\text{test}}, y_{\text{test}})$ with the disjoint label y_{test}, which verifies the result of knowledge transfer and adaptation.

3.2 Proposed framework

Figure 2 illustrates the framework of our proposed method. Two complementary classification tasks are performed simultaneously to learn the main encoder $\mathcal{F}_\theta(\cdot)$, which is a GNN for projecting a graph into an embedding. The first learning module is metric-based meta-learning, which utilizes explicit label information to generate the graph embedding and compute the similarity between the support set and query set. The second learning module is contrastive learning, which is a self-supervised instance-level classification task to improve the representation result. For self-supervised learning, we design a strategy to generate a pair of positive and negative augmentation views of the input graph automatically, which contributes to data copyright protection by mitigating the risk of unauthorized reproduction or misuse of the original data.

During meta-learning, the main encoder $\mathcal{F}_\theta(\cdot)$ maps each graph into a latent representation as its graph embedding $\boldsymbol{h}_{G_i} = \mathcal{F}_\theta(G_i)$. Specifically, GNNs compute graph embedding via a message-passing framework:

$$\boldsymbol{h}_u^{(l+1)} = \text{COM}\left(\boldsymbol{h}_u^{(l)}, \left[\text{AGG}\left(\left\{\boldsymbol{h}_{u'}^{(l)} | \forall u' \in \boldsymbol{U}'\right\}\right)\right]\right) \quad (1)$$

$$\boldsymbol{h}_G = \text{READOUT}\left(\boldsymbol{h}_u^{(l)} | \forall u \in \boldsymbol{U}\right) \quad (2)$$

where $\boldsymbol{h}_u^{(l)}$ denotes the embedding of node u at the l-th GNN layer; $\boldsymbol{U}'$ is the neighbor set of node u; AGG$(\cdot)$ is the neighbor aggregation function; COM$(\cdot)$ is the combination function; and READOUT$(\cdot)$ is the graph-level pooling function. Then all support graph embeddings in the same class y_n are aggregated into one

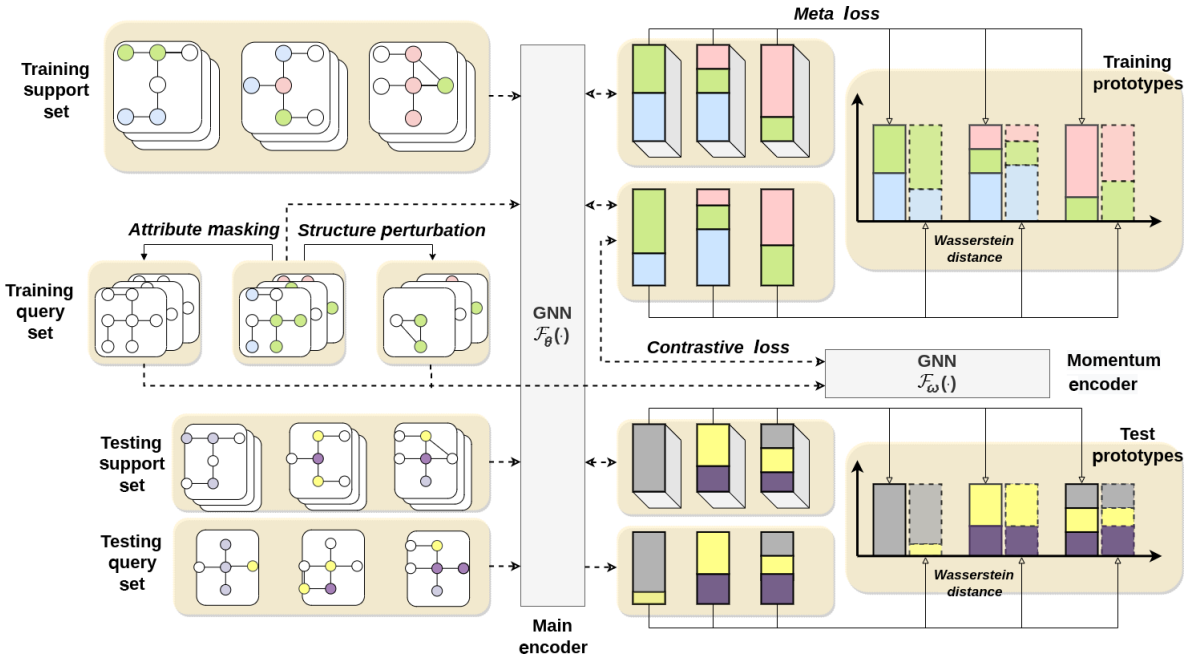


Fig. 2 Overview of SE-GCL. The framework consists of two main processes: graph meta-learning and contrastive learning. Given a support set of input graphs, we use a graph encoder to extract robust feature representation and derive reliable prototypes for each class. The Wasserstein metric measures the similarity between the query graph and the prototype. Further, we impose the contrastive loss on the query set to improve the model's generalizability. The complete workflow of all modules is an end-to-end solution. More details could be in the section of the proposed framework.

prototype representation z_n by computing the average, which is formulated as

$$z_n = \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{h}_{G_i} \ (G_i \in \boldsymbol{D}_S(G, y_n), n \in [1, N]) \quad (3)$$

To predict the label of the query graph, the similarity between query graph embedding and the prototype representation is measured by the p-th Wasserstein distance following the work in Ref. [36], which is the optional cost of moving mass between two graph embeddings. The classification loss $\mathcal{L}_{\text{Meta}}$ is defined as the average cross entropy between true labels and predictions based on the similarity, which can be formulated as

$$\mathcal{L}_{\text{Meta}}(\boldsymbol{D}_S, \boldsymbol{D}_Q, \theta) = -\frac{1}{M}\sum_{(G,y)\in \boldsymbol{D}_Q} \log \frac{e^{\text{sim}(\mathcal{F}_\theta(G),\, z_y)}}{\sum_{i=1}^{N} e^{\text{sim}(\mathcal{F}_\theta(G),\, z_i)}} \quad (4)$$

where sim denotes the Wasserstein similarity metric.

Because contrastive learning can maximize the agreement between the input data and its positive view while minimizing the agreement with the negative view, two automatic augmentations are employed to generate a pair of differentiable views for the respective goals, which reduces the need to unauthorized operations of the original data. Expressly, the positive augmentation operation preserves the original topology of the sample graph G_i and masks all the node features to form a positive view G_i^{mask}, which aims to mediate the overwhelming of the node features over the graph structure information in the representation learning. On the other hand, the negative augmentation operation generates a negative view G_i^{neg} by random node-dropping and edge-perturbation. Both operations follow an i.i.d. uniform distribution with node-dropping ratio η and edge-perturbation ratio $1-\eta$. For edge-perturbation, it randomly drops $1-\eta$ existing edges, then adds the same amount of random edges back into G_i. To form G_i^{neg} as a small subgraph from G_i with a few noisy edges, η is set to 0.8 by default. Moreover, it is stated in Ref. [23] that the structural information of graph data consists of both local and global dimensions, which means some attributes of a graph depend on the substructure of the graph while some consider the global structure more. As generalization is the main challenge for meta-learning to test novel domains, randomly treating a small subgraph as the negative example helps predictive models generalize beyond the limited training data. It should be noted that the negative view of one sample graph is also treated as the negative view of the rest

samples (i.e., for a query set containing M samples, there are M negative views for each sample graph). Introduced in Ref. [37], we apply a momentum encoder $\mathcal{F}_\omega(\cdot)$ for projecting the contrastive views, which behaves similarly as the main encoder as its parameter ω is a moving average of θ. Given $\mathcal{F}_\theta(G_i)$, the contrastive loss aims to maximize its agreement with $\mathcal{F}_\omega(G_i^{\text{mask}})$ while minimizing the agreement with all the negative views $\mathcal{F}_\omega(G_j^{\text{neg}})$, $j \in M$, which can be formulated as

$$\mathcal{L}_{\text{con}}(\boldsymbol{D}_S, \boldsymbol{D}_Q, \theta, \omega, \eta) = -\frac{1}{M}\sum_{G\in\boldsymbol{D}_Q}\log\frac{\mathrm{e}^{\text{sim}(\mathcal{F}_\theta(G),\mathcal{F}_\omega(G^{\text{mask}}))}}{\sum_{j=1}^{M}\mathrm{e}^{\text{sim}(\mathcal{F}_\theta(G),\mathcal{F}_\omega(G_j^{\text{neg}}))}} \tag{5}$$

where M denotes the size of the query set, and η is the perturbation ratio. By minimizing $\mathcal{L}_{\text{con}}$ w.r.t. θ, we force the main encoder $\mathcal{F}_\theta(\cdot)$ to maintain the complete structural information in the embedding and produce more generalized prototypical networks. Thus, the overall loss is the combination of the classification loss and the contrastive loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Meta}}(\boldsymbol{D}_S, \boldsymbol{D}_Q, \theta) + \beta\mathcal{L}_{\text{con}}(\boldsymbol{D}_S, \boldsymbol{D}_Q, \theta, \omega, \eta) \tag{6}$$

where β is a hyper-parameter that balances two terms. The detailed learning process is described in Algorithm 1. And all notations used in this paper are listed in Table 1.

4 Experiment

In this section, we present the experiments developed by PyTorch Geometric and conducted on a workstation with an Intel Core i7 2.80 GHz CPU and an NVIDIA GeForce GTX 1070 GPU. Our proposed method is evaluated on standard few-shot learning benchmarks with real-world datasets. We also conduct an ablative study about the effectiveness of the contrastive learning module in our framework.

Table 1 List of notations used in this paper.

Symbol	Description
G	Undirected unweighted graph
$\boldsymbol{U}$	Set of nodes
$\boldsymbol{U}'$	Set of node's neighbors
$\boldsymbol{E}$	Set of edges
$\boldsymbol{A}$	Set of node attributes
y	Graph label
$\boldsymbol{D}_S$	Support dataset
$\boldsymbol{D}_Q$	Query dataset
$\mathcal{F}_\theta(\cdot)$	Main graph encoder
$\mathcal{F}_\omega(\cdot)$	Momentum graph encoder
$\boldsymbol{h}_G$	Graph embedding
$\boldsymbol{h}_u^{(l+1)}$	Node embedding at the l-th GNN layer
$\boldsymbol{z}_n$	Graph prototype representation
G^{mask}	Graph positive augmentation view
G^{neg}	Graph negative augmentation view
η	Perturbation ratio of G^{neg}
β	Regularization hyper-parameter

4.1 Dataset

In the experiments, we use a variety of large and small attributed networks that are collected from different domains, including e-commerce networks, citation networks, and morphological networks. It is worth noting that meta-learning tasks often demand a considerable number of classes, whereas some commonly employed graph datasets[38] have limited classes or small node-attribute dimensions. Therefore, to validate the effectiveness of the proposed framework, we construct graph datasets from two large attributed networks with lots of node classes, Amazon-Clothing and DBLP, and use the node-label distribution as the label of the new graph. Meanwhile, we adapt two small attributed networks Letter-High and TRIANGLES, which are used in GSM[22] for a fair comparison. To ensure that appropriate data usage agreements are in place with the data owners or providers, we specify

Algorithm 1 Learning process of SE-GCL

Input: Graph dataset: $\{G_1, G_2, \ldots, G_m\}$,
graphs' labels: $\{y_1, y_2, \ldots, y_m\}$,
task: $\mathcal{T}_{\text{test}}$=$\{\boldsymbol{D}_S(G_{\text{test}}, y_{\text{test}}), \boldsymbol{D}_Q(G_{\text{test}}, y_{\text{test}})\}$,
training episodes: T, perturbation ratio: η,
learning rate: α, momentum coefficient: ϵ.
Output: Predicted labels of G_{test} in $\boldsymbol{D}_Q$

while $i < T$ **do**
  //Meta-training process
  Sample a meta-training task:
  $\mathcal{T}_{\text{train}}^i$=$\{\boldsymbol{D}_S(G_{\text{train}}, y_{\text{train}}), \boldsymbol{D}_Q(G_{\text{train}}, y_{\text{train}})\}$;
  Compute the prototype representations $\boldsymbol{z}_{\text{train}}$ of support set $\boldsymbol{D}_S(G_{\text{train}})$ according to Eq. (3);
  //Contrastive process
  Generate the augmentation views G^{mask} and G^{neg} of $\boldsymbol{D}_Q(G_{\text{train}})$ with η;
  Update the main encoder by minimizing loss in Eq. (6):
  $\theta^{i+1} = \theta^i - \alpha\nabla_{\theta^i}\mathcal{L}_{\text{total}}$;
  Update the momentum encoder with ϵ:
  $\omega^{i+1} = \epsilon\omega^i + (1-\epsilon)\theta^{i+1}$;
  //Meta-testing process
  Compute the prototype representations $\boldsymbol{z}_{\text{test}}$ of support set $\boldsymbol{D}_S(G_{\text{test}})$ from $\mathcal{T}_{\text{test}}$ according to Eq. (3);
  Predict the labels of $\boldsymbol{D}_Q(G_{\text{test}})$ from $\mathcal{T}_{\text{test}}$ using the prototypical networks.
end while
return Predicted labels of G_{test} in $\boldsymbol{D}_Q$

the scope and limitations of data usage. The detailed descriptions of these datasets are as follows:

Amazon-Clothing. The dataset was originally collected by Ref. [39] and has been preprocessed by Ref. [12] for the FSL study. In our constructed dataset, each graph represents a customer's shopping history, where each node corresponds to a product, and different products are connected if the same customer browses them. The product descriptions are used as node attributes. We customize 2000 graphs with 20 types of shopping habits from 77 kinds of products for FSL.

DBLP. The citation network is extracted from Ref. [40] with node features generated by Ref. [41] using the Bag-of-Words model. For this dataset, we follow the same construction method to customize 2000 graphs with 20 graph classes, where each node represents a paper and edges represent citations.

Letter-High. Each graph is a distorted alphabetic prototype graph with undirected edges and vertices representing lines and ending points of lines[42]. More specifically, Letter-High contains 15 categories from the English alphabet: A, E, F, H, I, K, L, M, N, T, V, W, X, Y, and Z.

TRIANGLES. This dataset contains 10 different graph classes numbered from 1 to 10, corresponding to the number of triangles in the graphs of each class. The partial version is used in the experiments in Ref. [22] that reduces the graph sample size from 45 000 to 2000.

All statistic information of these datasets are listed in Table 2.

4.2 Baselines and implementation

We compare our method with the following five types of baselines:

Weisfeiler-Lehman graph kernels[43], based on the Weisfeiler-Lehman (WL) test of graph isomorphism, is considered as the state-of-the-art in graph classification.

Table 2 Statistics of datasets. We show each dataset with the number of graphs $|G|$, the average number of nodes Avg.$|U|$, the average number of edges Avg.$|E|$, the dimensions of node attributes $|A|$, and the number of classes for training over testing $|y_{\text{train}}|/|y_{\text{test}}|$.

Dataset	$\lvert G\rvert$	Avg.$\lvert U\rvert$	Avg.$\lvert E\rvert$	$\lvert A\rvert$	$\frac{\lvert y_{\text{train}}\rvert}{\lvert y_{\text{test}}\rvert}$
Amazon-Clothing	2000	32.15	192.50	9034	10/10
DBLP	2000	47.25	318.45	7202	10/10
Letter-High	2250	4.67	4.50	2	11/4
TRIANGLES	2000	20.85	35.50	1	7/3

We skip the unsuitable meta-training phase for this method and perform N-way-K-shot graph classification directly on the testing dataset.

GIN[24] uses injective neighbor aggregation to approximately conceive through WL test, which considers performing better than GCN and GraphSAGE[44] in case of graph classification. Thus, we train a naive GIN + MLP classifier directly on the testing dataset to verify the knowledge transfer ability in meta-learning.

MAML[20] is an optimization-based meta-learning method that tries to learn better model initialization from a series of meta-training tasks. For the few-shot graph classification task, we extend it by the same graph encoder backbone in our framework.

GSM[22] clusters the graph classes based on the graph spectral measures into groups named super-classes and uses the constructed super-classes for few-shot learning.

PN has the identical architecture of our framework but without the contrastive learning module, which can be considered a variant of a prototypical network.

For either baselines or our framework, we implement the graph encoder consisting of three GCN layers[45] or GAT layers[46] with dimension sizes 32, 32, and 16, respectively. All the layers are activated with the ReLU function. We choose Mincut pooling[47] as the readout operation because it coarsens a graph by taking into account both the connectivity structure and the node features. The loss is trained with Adam optimizer, whose learning rate is set to 0.001 initially with a weight decay of 0.0001. We adjust the dropout rate and the perturbation ratio η for each dataset to achieve the best performance, and train the model with an early-stopping strategy across 300 episodes. Moreover, we set the regularization hyperparameter β to 1.0 and the momentum coefficient ϵ to 0.99 by default.

4.3 Result analysis

We evaluate the performance by 5-way-5-shot, 5-way-10-shot, and 8-way-5-shot tasks on both Amazon-Clothing and DBLP datasets, whose metrics Accuracy (ACC) results are presented in Table 3. For Letter-High and TRIANGLES containing few node information, we perform 4-way and 3-way tasks, respectively, and only compare them with GSM's best results in Table 4. From a comprehensive view, we have the following observations:

(1) For WL kernel and GIN baselines, we perform N-way-K-shot graph classification directly over the

Table 3 Accuracy with a standard deviation of baselines and our method. We tested 100 *N*-way-*K*-shot tasks on both Amazon-Clothing and DBLP datasets. The best results are highlighted in bold.

Method	Accuracy (%)					
	Amazon-Clothing			DBLP		
	5-way 5-shot	5-way 10-shot	8-way 5-shot	5-way 5-shot	5-way 10-shot	8-way 5-shot
WL kernel	56.40 ± 2.23	65.24 ± 1.37	49.47 ± 2.64	57.12 ± 2.44	65.52 ± 1.71	50.35 ± 2.39
GIN	63.25 ± 1.63	71.24 ± 1.57	55.47 ± 3.34	66.10 ± 2.41	72.38 ± 1.44	57.13 ± 2.88
MAML (GCN)	70.72 ± 3.88	76.62 ± 2.35	60.70 ± 4.53	73.12 ± 4.65	77.69 ± 2.89	63.19 ± 5.12
MAML (GAT)	70.66 ± 3.53	76.68 ± 2.51	60.27 ± 4.49	74.10 ± 4.19	78.03 ± 3.44	62.80 ± 3.99
PN (GCN)	70.18 ± 1.19	77.43 ± 1.87	63.17 ± 2.14	74.32 ± 2.49	79.79 ± 2.19	64.49 ± 3.19
PN (GAT)	71.22 ± 2.43	77.06 ± 2.15	63.89 ± 2.94	74.91 ± 3.29	80.29 ± 2.34	64.52 ± 3.52
SE-GCL (GCN)	74.98 ± 2.01	80.22 ± 1.55	66.37 ± 1.99	77.31 ± 2.17	83.40 ± 1.14	67.59 ± 2.86
SE-GCL (GAT)	**75.02 ± 2.90**	**81.76 ± 2.36**	**66.92 ± 2.43**	**78.16 ± 3.09**	**84.75 ± 1.82**	**68.25 ± 3.20**

Table 4 Accuracy of GSM and our method. We tested 100 *N*-way-*K*-shot tasks on both Letter-High (4-way) and TRIANGLES datasets (3-way).

Method	Accuracy (%)		
	K-shot	Letter-High	TRIANGLES
GSM	5	69.91 ± 5.90	71.40 ± 4.34
	10	73.28 ± 3.46	75.60 ± 3.67
	20	77.38 ± 1.58	80.04 ± 2.20
SE-GCL	5	74.34 ± 1.03	77.36 ± 1.25
	10	79.42 ± 0.84	83.14 ± 1.07
	20	84.15 ± 0.77	89.17 ± 0.85

test classes. It is clear that the sample size restricts the accuracy of the prediction results. When there are insufficient samples for training, the model is prone to overfit, which leads to unsatisfactory test results. GIN incorporates node features while generalizing the WL test, so the effect is better than the WL kernel, which shows that node features play a vital role in graph learning.

(2) Both MAML and PN achieve superior performances on three types of graph classification tasks over GIN, indicating that for Amazon-Clothing and DBLP datasets, meta-learning can improve the learning process of new tasks using the experience gained from solving predecessor problems. MAML and PN have similar performance on 5-way tasks, but PN obtains about 3.5% performance gains on 8-way tasks. The reason is twofold: as an optimization-based approach, the generalization ability of MAML is getting poor when the number of classification labels increases and the difference in sample data becomes large. The performance suffers from its fine-tuning process. While as a metric-based approach, PN learns generalizable matching metrics by taking the mean vector of support examples, which is simple but stable with favorable distance metrics.

(3) It is worth noting that SE-GCL extends the PN basis with the contrastive learning method, further enhancing the model's generalization ability. Forged by contrastive learning, the positive and negative samples strengthen the graph structure learning and make up for the insufficient number of few-shot samples to a certain extent. Overall, SE-GCL outperforms the baselines in all the tasks on both Amazon-Clothing and DBLP datasets. At the same time, we also find that as the encoder backbone, the gap between GCN and GAT is not apparent, which means that the aggregation function (i.e., the main difference in message passing between GCN and GAT) has much less impact on the graph-level than the pooling and readout operations.

(4) As shown in Table 4, SE-GCL outperforms GSM by 5% for both datasets, though both SE-GCL and GSM use metric-based approaches to solve the few-shot problem. This is because GSM assumes that the test classes could belong to the same super-classes built from the training classes. However, training and test classes typically do not overlap in the few-shot setting. On the other hand, we believe the ability of the GNN encoder to learn graph representation in a top-down way is more critical when encountering unseen classes, where the effectiveness of the Mincut pooling strategy on unsupervised node clustering helps in the simple-graph dataset with few nodes information. Moreover, SE-GCL can alleviate the overfitting problem caused by simple graph topology through contrastive learning; even SE-GCL is more suitable for graph datasets with complex topology and excessive node information. By incorporating such complexities into the graph data, it becomes more challenging for unauthorized individuals to extract or identify specific sensitive information from the copyrighted data.

(5) We visualize the DBLP dataset in 2-dimensional

space by applying the t-SNE algorithm to the graph embeddings, which are learned from 5-way-10-shot classification tasks using the baselines and our method. The results shown in Fig. 3 validate our assertions in the previous section through a meaningful layout. We can see that our method creates better clusters with low intra-cluster and high inter-cluster distances.

4.4 Parameter analysis

In this section, extensive experiments are conducted to analyze the influence of the perturbation ratio η to SE-GCL, whose results are shown in Fig. 4. We review the negative augmentation operation as a combination of node-dropping and edge-perturbation with ratios η and 1-η, respectively. A small η renders a negative view having similar nodes in the sample graph but connected by different edges, while a hefty η generates a subgraph of the sample graph with a few random edges. According to Fig. 4, the performances are better with large values of η in datasets DBLP, TRIANGLES, and Amazon-Clothing, while the performance deteriorates in dataset Letter-High with η increasing. There are two main reasons behind this phenomenon: First, the graphs of Letter-High are sparse with a moderately low number of nodes. With a large node-dropping ratio, the negative view may only have 1 or 2 nodes, which cannot provide valuable information for contrastive learning. Moreover, these small attributed graphs are more sensitive to individual edges, leading edge-perturbation to generate meaningful negative views. Second, the graphs of DBLP, TRIANGLES, and Amazon-Clothing have a large number of nodes with complex topology. Negative views generated by extensive edge-perturbation disturb the contrastive module due to being incompatible with node attributes and are empirically unhelpful for downstream performance. In contrast, node-dropping and subgraphs are beneficial across datasets by enforcing the consistency and generality of local sample graphs and global prototypes.

5 Conclusion

In this study, we address a practical issue in few-shot graph classification concerning the copyright protection of graph data. We observe an imbalance in the utilization

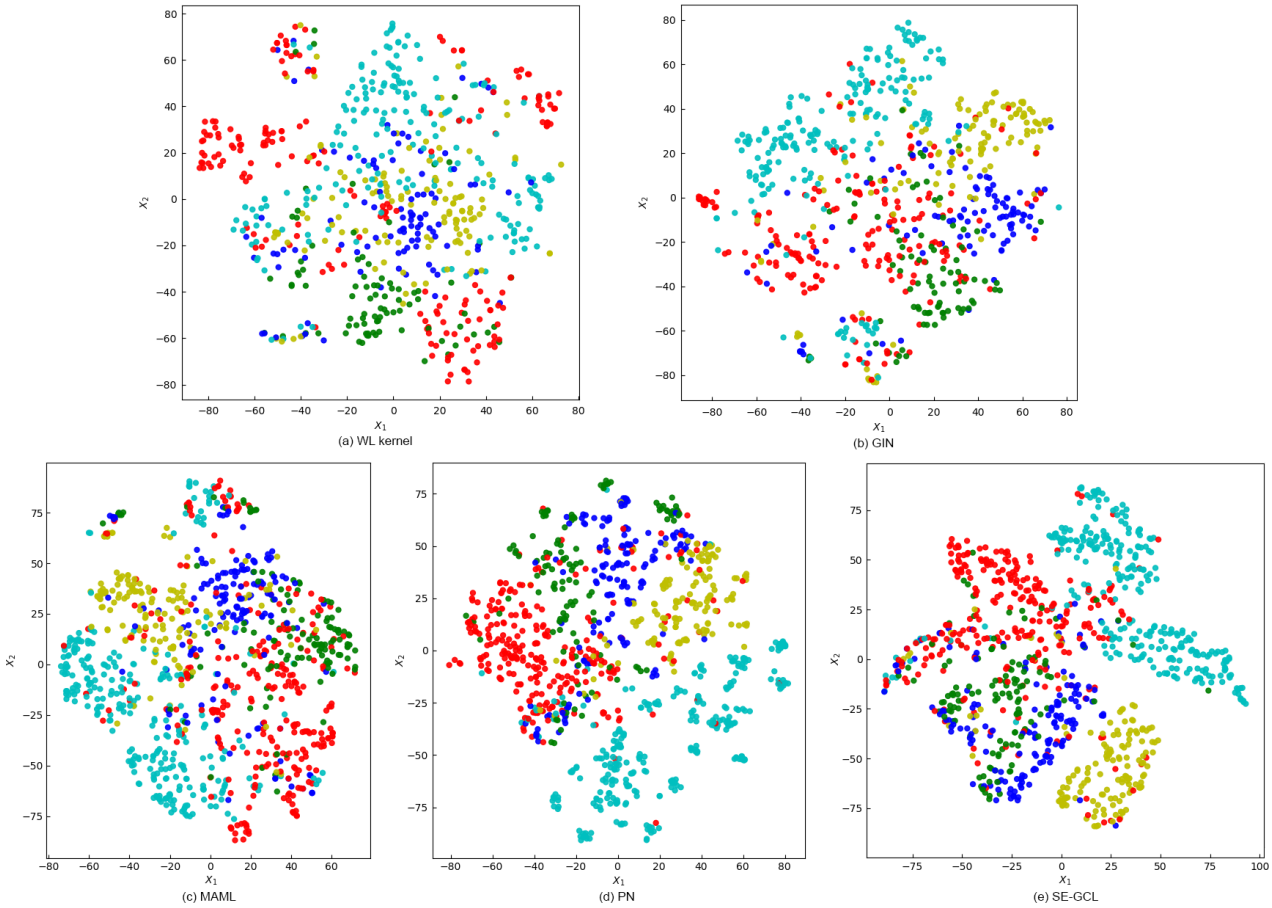


Fig. 3 *t*-SNE visualization comparison for the DBLP dataset. Each class is represented in a different color.

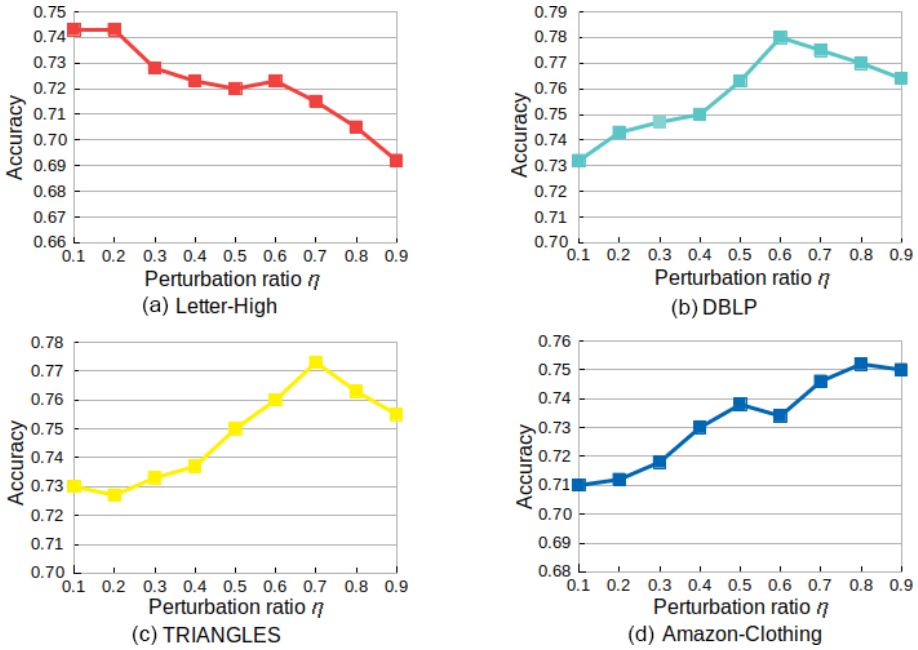


Fig. 4 Influence of the perturbation ratio η. The range of η is set from 0.1 to 0.9.

of node attributes and graph structure during the learning process and propose a novel meta-learning framework with a contrastive learning module to enhance the learning of graph structure. On one hand, the prototype networks based on the Wasserstein similarity metric allow the uncertainty distribution to encompass task embeddings beyond the training set, which enables the model to generalize to unseen test tasks after meta-training. On the other hand, the contrastive module introduces meaningful positive and negative views, which regularize the model to prioritize the global structure of the graph over partial node attributes or subgraph features. The experimental results demonstrate that our framework achieves outstanding performance compared to other baselines, whether applied to large or small attributed graph datasets. As a future direction, we aim to develop automatic augmentation strategies within the contrastive learning module to prevent unauthorized use of original works and copyright infringement. By defining the objectives and parameters of data augmentation, organizations can exercise control over synthetic data to ensure compliance with copyright and privacy regulations.

Acknowledgment

This work was supported by SW Copyright Ecosystem R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports, and Tourism in 2023 (No. RS-2023-00224818).

References

[1] X. Zheng and Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial IoTs, *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, 2020.

[2] Z. Cai and X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 766–775, 2020.

[3] Z. Cai, X. Zheng, J. Wang, and Z. He, Private data trading towards range counting queries in Internet of Things, *IEEE Trans. Mob. Comput.*, vol. 22, no. 8, pp. 4881–4897, 2023.

[4] Z. Cai, Z. He, X. Guan, and Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 577–590, 2018.

[5] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, Deep learning based inference of private information using embedded sensors in smart devices, *IEEE Netw.*, vol. 32, no. 4, pp. 8–14, 2018.

[6] Y. Huang, Y. J. Li, and Z. Cai, Security and privacy in metaverse: A comprehensive survey, *Big Data Mining and Analytics*, vol. 6, no. 2, pp. 234–247, 2023.

[7] I. K. Nti, J. A. Quarcoo, J. Aning, and G. K. Fosu, A mini-review of machine learning in big data analytics: Applications, challenges, and prospects, *Big Data Mining and Analytics*, vol. 5, no. 2, pp. 81–97, 2022.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[9] B. M. Oloulade, J. Gao, J. Chen, T. Lyu, and R. Al-Sabri, Graph neural architecture search: A survey, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 692–708, 2022.

[10] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, An end-to-

end deep learning architecture for graph classification, *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, pp. 4438–4445, 2018.

[11] Y. Duan, J. Wang, H. Ma, and Y. Sun, Residual convolutional graph neural network with subgraph attention pooling, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 653–663, 2022.

[12] F. Errica, M. Podda, D. Bacciu, and A. Micheli, A fair comparison of graph neural networks for graph classification, arXiv preprint arXiv: 1912.09893, 2019.

[13] H. Zhao, H. Chen, L. Li, and H. Wan, Understanding social relationships with person-pair relations, *Big Data Mining and Analytics*, vol. 5, no. 2, pp. 120–129, 2022.

[14] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec, Pixie: A system for recommending 3+ billion items to 200+ million users in real-time, in *Proc. 2018 World Wide Web Conference*, 2018, pp. 1775–1784.

[15] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, *ACM Comput. Surv.*, vol. 53, no. 3, p. 63, 2020.

[16] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, Meta-baseline: Exploring simple meta-learning for few-shot learning, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Montreal, Canada, 2021, pp. 9042–9051.

[17] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, Meta-learning in neural networks: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, 2021.

[18] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, Matching networks for one shot learning, in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 3637–3645.

[19] J. Snell, K. Swersky, and R. Zemel, Prototypical networks for few-shot learning, in *Proc. 31st Int. Conf. Neural Information Processing Systems*, 2017, pp. 4080–4090.

[20] C. Finn, P. Abbeel, and S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in *Proc. 34th Int. Conf. Machine Learning - Volume 70*, Sydney, Australia, 2017, pp. 1126–1135.

[21] N. Ma, J. Bu, J. Yang, Z. Zhang, C. Yao, Z. Yu, S. Zhou, and X. Yan, Adaptivestep graph meta-learner for few-shot graph classification, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020, pp. 1055–1064.

[22] J. Chauhan, D. Nathani, and M. Kaul, Few-shot learning on graphs via super-classes based on graph spectral measures, arXiv preprint arXiv: 2002.12815, 2020.

[23] S. Jiang, F. Feng, W. Chen, X. Li, and X. He, Structure-enhanced meta-learning for few-shot graph classification, *AI Open*, vol. 2, pp. 160–167, 2021.

[24] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, How powerful are graph neural networks? arXiv preprint arXiv: 1810.00826v1, 2019.

[25] K. Hassani, Cross-domain few-shot graph classification, arXiv preprint arXiv: 2201.08265, 2022.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, A simple framework for contrastive learning of visual representations, in *Proc. 37th Int. Conf. Machine Learning*, Vienna, Austria, 2020, pp. 1597–1607.

[27] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, Graph contrastive learning with augmentations, in *Proc. 34th Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, pp. 5812–5823.

[28] Y. You, T. Chen, Y. Shen, and Z. Wang, Graph contrastive learning automated, arXiv preprint arXiv: 2106.07594, 2021.

[29] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, Graph contrastive learning with adaptive augmentation, in *Proc. Web Conf. 2021*, Ljubljana, Slovenia, 2021, pp. 2069–2080.

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

[31] D. Xu, W. Cheng, D. Luo, H. Chen, and X. Zhang, InfoGCL: Information-aware graph contrastive learning, arXiv preprint arXiv: 2110.15438, 2021.

[32] S. Suresh, P. Li, C. Hao, and J. Neville, Adversarial graph augmentation to improve graph contrastive learning, arXiv preprint arXiv: 2106.05819v4, 2021.

[33] S. Lin, C. Liu, P. Zhou, Z. Y. Hu, S. Wang, R. Zhao, Y. Zheng, L. Lin, E. Xing, and X. Liang, Prototypical graph contrastive learning, *IEEE Trans. Neural Netw. Learn. Syst.*, doi: 10.1109/TNNLS.2022.3191086.

[34] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, Are graph augmentations necessary? Simple graph contrastive learning for recommendation, in *Proc. 45th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Madrid, Spain, 2022, pp. 1294–1303.

[35] Y. Zhu, Y. Xu, Q. Liu, and S. Wu, An empirical study of graph contrastive learning, arXiv preprint arXiv: 2109.01116v2, 2021.

[36] S. Kolouri, N. Naderializadeh, G. K. Rohde, and H. Hoffmann, Wasserstein embedding for graph learning, arXiv preprint arXiv: 2006.09430, 2020.

[37] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, Momentum contrast for unsupervised visual representation learning, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 9726–9735.

[38] K. Kersting, N. M. Kriege, C. Morris, P. Mutzel, and M. Neumann, Benchmark data sets for graph kernels, https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets, 2016.

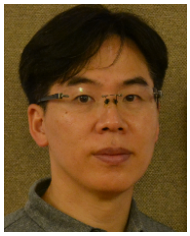
[39] J. McAuley, R. Pandey, and J. Leskovec, Inferring networks of substitutable and complementary products, in *Proc. 21th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Sydney, Australia, 2015, pp. 785–794.

[40] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, ArnetMiner: Extraction and mining of academic social networks, in *Proc. 14th ACM SIGKDD Int. Conf. Knowledge discovery and data mining*, Las Vegas, NV, USA, 2008, pp. 990–998.

[41] N. Wang, M. Luo, K. Ding, L. Zhang, J. Li, and Q. Zheng, Graph few-shot learning with attribute matching, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020, pp. 1545–1554.

[42] K. Riesen and H. Bunke, IAM graph database repository for graph based pattern recognition and machine learning, https://doi.org/10.1007/978-3-540-89689-0_33, 2008.
[43] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, Weisfeiler-lehman graph kernels, *J. Mach. Learn. Res.*, vol. 12, pp. 2539–2561, 2011.
[44] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, Inductive representation learning on temporal graphs, arXiv preprint arXiv: 2002.07962v1, 2020.
[45] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv: 1609.02907, 2016.
[46] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, Graph attention networks, arXiv preprint arXiv: 1710.10903v3, 2018.
[47] F. M. Bianchi, D. Grattarola, and C. Alippi, Mincut pooling in graph neural networks, arXiv preprint arXiv: 1907.00481v2, 2020.

Kainan Zhang received the BS degree in computer science from Zhejiang University of Technology, Hangzhou, China, and the MS degree in computer engineering from Illinois Institute of Technology, Chicago, IL, USA. He is currently pursuing the PhD degree at Department of Computer Science, Georgia State University, Atlanta, GA, USA. His research interests include privacy preservation, social networking, and machine learning.



DongMyung Shin received the BS, MS, and PhD degrees in computer engineering from Daejeon University, Republic of Korea in 1997, 2000, and 2003, respectively. In 2001, he joined Korea Information Security Agency (KISA), where he served as a senior researcher in the Applied Security Technology Team and later in the Software Appraisal Team at Computer Program Deliberation & Mediation Committee (PDMC). In 2009, he conducted research on contents filtering of copyright works at Korea Copyright Committee (KCC), and in 2014, he focused on smart grid security at Korea Smart Grid Institute (KSGI). Currently, he is working as the CTO at LSWare Inc, Republic of Korea. His research interests include system and network security, blockchain development, various Internet Communication Technology (ICT) convergence, and open-source software license checks.

Daehee Seo received the BS degree in electronic and electrical engineering from Dongshin University, Naju, Republic of Korea in 2001, and the MS degree in computer science and engineering and the PhD degree in computer science from Soonchunhyang University, Republic of Korea in 2003 and 2006, respectively. Currently, he serves as an assistant professor at College of Intelligence Information Engineering, Sangmyung University (SMU), Seoul, Republic of Korea. He is a member of IEEE.

Zhipeng Cai received the PhD and MS degrees from University of Alberta, Canada, in 2008 and 2004, respectively, and the BS degree from Beijing Institute of Technology, China. He is currently a professor at Department of Computer Science, Georgia State University, USA, and also an affiliate professor at Department of Computer Information System, Robinson College of Business, as well as an associate director at INSPIRE Center. His research expertise lies in the areas of resource management and scheduling, high performance computing, privacy, networking, and big data. He is the recipient of an NSF CAREER Award.