Augmentation-Free Self-Supervised Learning on Graphs

Namkyeong Lee[1], **Junseok Lee**[1], **Chanyoung Park**[1,2*]

[1] Dept. of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea
[2] Graduate School of Artificial Intelligence, KAIST, Daejeon, Republic of Korea
namkyeong96@kaist.ac.kr, junseoklee@kaist.ac.kr, cy.park@kaist.ac.kr

Abstract

Inspired by the recent success of self-supervised methods applied on images, self-supervised learning on graph structured data has seen rapid growth especially centered on augmentation-based contrastive methods. However, we argue that without carefully designed augmentation techniques, augmentations on graphs may behave arbitrarily in that the underlying semantics of graphs can drastically change. As a consequence, the performance of existing augmentation-based methods is highly dependent on the choice of augmentation scheme, i.e., hyperparameters associated with augmentations. In this paper, we propose a novel augmentation-free self-supervised learning framework for graphs, named AFGRL. Specifically, we generate an alternative view of a graph by discovering nodes that share the local structural information and the global semantics with the graph. Extensive experiments towards various node-level tasks, i.e., node classification, clustering, and similarity search on various real-world datasets demonstrate the superiority of AFGRL. The source code for AFGRL is available at https://github.com/Namkyeong/AFGRL.

Introduction

Recently, self-supervised learning paradigm (Liu et al. 2021), which learns representation from supervision signals derived from the data itself without relying on human-provided labels, achieved great success in many domains including computer vision (Gidaris, Singh, and Komodakis 2018; Noroozi and Favaro 2016), signal processing (Banville et al. 2021, 2019), and natural language processing (Devlin et al. 2018; Brown et al. 2020). Specifically, contrastive methods, which are at the core of self-supervised learning paradigm, aim to build effective representation by pulling semantically similar (positive) pairs together and pushing dissimilar (negative) pairs apart (Hjelm et al. 2018; Oord, Li, and Vinyals 2018), where two augmented versions of an image are considered as positives, and the remaining images are considered as negatives. Inspired by the success of the contrastive methods in computer vision applied on images, these methods have been recently adopted to graphs (Xie et al. 2021).

*Corresponding author.

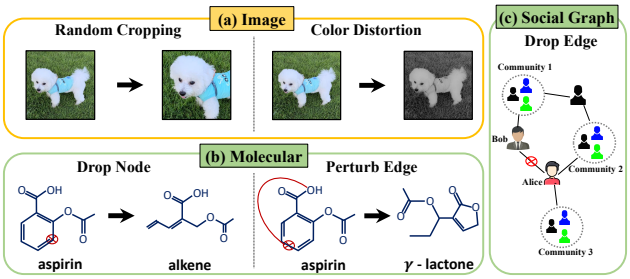


Figure 1: Augmentations on images ((a)) keep the underlying semantics, whereas augmentations on graphs ((b),(c)) may unexpectedly change the semantics.

Although self-supervised contrastive methods have been shown to be effective on various graph-related tasks, they pay little attention to the inherent distinction between images and graphs: *while augmentation is well defined on images, it may behave arbitrarily on graphs.* For example, in the case of images, even after randomly cropping and rotating them, or distorting their color, their underlying semantic is hardly changed (Figure 1 (a)), and even if the semantic changes, humans can readily recognize the change visually, and choose an alternative augmentation approach that preserves the semantic. On the other hand, when we perturb (drop or add) edges/nodes, and their features of a graph, we cannot ascertain whether the augmented graph would be positively related to the original graph, and what is worse, it is non-trivial to verify the validity of the augmented graph since graphs are hard to visualize. For example, in molecular graphs, dropping a carbon atom from the phenyl ring of aspirin breaks the aromatic system and results in a alkene chain (Figure 1(b)). Moreover, perturbing the connection of aspirin might introduce a molecule of totally different property, namely, five-membered lactone (Sun et al. 2021a). Likewise, in social graphs, randomly dropping edges might also lead to semantic changes, especially when these edges are related to hub nodes. For example, as shown in Figure 1(c), if the edge between Bob and Alice is dropped, it would take much longer distance for Bob to reach Community 3 (i.e., from 2-hops to 5-hops), which also alters the relationship between Community 1 and Community 3. We argue that this is mainly because graphs contain not only the

semantic but also the *structural information*.

Due to the aforementioned arbitrary behavior of augmentation on graphs, the quality of the learned graph representations of previous augmentation-based contrastive methods (Hassani and Khasahmadi 2020; Zhu et al. 2020, 2021; Thakoor et al. 2021; Sun et al. 2021a; Veličković et al. 2018) is highly **dependent on the choice of the augmentation scheme**. More precisely, in order to augment graphs, these methods perform various augmentation techniques such as node/edge perturbation and node feature masking, and the amount by which the graphs are augmented is controlled by a set of hyperparameters. However, these hyperparameters should be carefully tuned according to which datasets, and which downstream tasks are used for the model evaluation, otherwise the model performance would vary greatly (You et al. 2021). Moreover, it is also shown that the performance on downstream tasks highly resort to which combinations of the augmentation techniques (You et al. 2020; Sun et al. 2021a) are used.

Furthermore, even after discovering the best hyperparameters for augmentations, another limitation arises due to the inherent philosophy of contrastive learning. More precisely, inheriting the principle of instance discrimination (Wu et al. 2018), contrastive methods treat two samples as a positive pair as long as they are two augmented versions of the same instance, and all other pairs are treated as negatives. Although this approach is effective for learning representations of images (Chen et al. 2020; He et al. 2020), simply adopting it to graphs by **treating all other nodes apart from the node itself as negatives overlooks the structural information** of graphs, and thus cannot benefit from the relational inductive bias of graph-structured data (Battaglia et al. 2018). Lastly, due to the nature of contrastive methods, **a large amount of negative samples** is required for improving the performance on the downstream tasks, requiring high computational and memory costs, which is impractical in reality (Thakoor et al. 2021).

Contribution . We propose a self-supervised learning framework for graphs, called Augmentation-Free Graph Representation Learning (AFGRL), which *requires neither augmentation techniques nor negative samples* for learning representations of graphs. Precisely, instead of creating two arbitrarily augmented views of a graph and expecting them to preserve the semantics of the original graph, we use the original graph per se as one view, and generate another view by discovering, for each node in the original graph, nodes that can serve as *positive samples* via k-nearest-neighbor (k-NN) search in the representation space. Then, given the two semantically related views, we aim to predict, for each node in the first view, the latent representations of its positive nodes in the second view. However, naively selecting positive samples based on k-NN search to generate an alternative view can still alter the semantics of the original graph.

Hence, we introduce a mechanism to filter out false positives from the samples discovered by k-NN search. In a nutshell, we consider a sample to be positive only if either 1) it is a neighboring node of the target node in the adjacency matrix (local perspective), capturing the relational inductive bias inherent in the graph-structured data, or 2) it belongs to the same cluster as the target node (global perspective). Moreover, by adopting BYOL (Grill et al. 2020) as the backbone of our model, negative samples are not required for the model training, thereby avoiding the *"sampling bias"* (Lin et al. 2021), i.e. the negative samples may have the same semantics with the query node, which would result in less effective representation (Saunshi et al. 2019).

Our extensive experiments demonstrate that AFGRL outperforms a wide range of state-of-the-art methods in terms of node classification, clustering and similarity search. We also demonstrate that compared with existing methods that heavily depend on the choice of hyperparameters, AFGRL is stable over hyperparameters. To the best of our knowledge, AFGRL is the first work that learns representations of graphs without relying on manual augmentation techniques and negative samples.

Related Work

Recently, motivated by the great success of self-supervised methods on images, contrastive methods have been increasingly adopted to graphs. DGI (Veličković et al. 2018), a pioneering work highly inspired by Deep InfoMax (Hjelm et al. 2018), aims to learn node representations by maximizing the mutual information between the local patch of a graph. i.e., node, and the global summary of the graph, thereby capturing the global information of the graph that is overlooked by vanilla graph convolutional networks (GCNs) (Kipf and Welling 2016; Veličković et al. 2017). DGI is further improved by taking into account the mutual information regarding the edges (Peng et al. 2020) and node attributes (Jing, Park, and Tong 2021). Inspired by SimCLR (Chen et al. 2020), GRACE (Zhu et al. 2020) first creates two *augmented views* of a graph by randomly perturbing nodes/edges and their features. Then, it learns node representations by pulling together the representation of the same node in the two augmented graphs, while pushing apart representations of every other node. This principle (Wu et al. 2018) has also been adopted for learning graph-level representations of graphs that can be used for graph classification, (Sun et al. 2019; You et al. 2020; Hassani and Khasahmadi 2020). Despite the success of contrastive methods on graphs, they are criticized for the problem raised by the "sampling bias" (Bielak, Kajdanowicz, and Chawla 2021). Moreover, these methods require a large amount of negative samples for the model training, which incurs high computational and memory costs (Grill et al. 2020).

To address the sampling bias issue, BGRL (Thakoor et al. 2021) learns node representations without using negative samples. It learns node representations by encoding two augmented versions of a graph using two separate encoders: one is trained through minimizing the cosine loss between the representations generated by the two encoders, while the other encoder is updated by an exponential moving average of the first encoder. Although the sampling bias has been addressed in this way, BGRL still requires augmentations on the original graph, which may lead to semantic drift (Sun et al. 2021a) as illustrated in Figure 1. On the other hand,

our proposed method learns node representations without any use of negative samples or augmentations of graphs.

Augmentations on Graphs. Most recently, various augmentation techniques for graphs have been introduced. e.g., node dropping (You et al. 2020), edge modification (Jin et al. 2021; Qiu et al. 2020; Zhao et al. 2020), subgraph extraction (Jiao et al. 2020; Sun et al. 2021b), attribute masking (Zhu et al. 2020, 2021) and others (Hassani and Khasahmadi 2020; Kefato and Girdzijauskas 2021; Suresh et al. 2021). GRACE (Zhu et al. 2020) randomly drops edges and masks node features to generate two augmented views of a graph. GCA (Zhu et al. 2021) further improves GRACE by introducing advanced adaptive augmentation techniques that take into account both structural and attribute information. However, due to the complex nature of graphs, the performance on downstream tasks is highly dependent on the selection of the augmentation scheme, as will be shown later in our experiments (Table 1). Moreover, previous work (You et al. 2020; Sun et al. 2021a) have shown that there is no universally outperforming data augmentation scheme for graphs. Lastly, Sun et al. (2021a) demonstrates that infusing domain knowledge is helpful in finding proper augmentations, which preserves biological assumption in molecular graph. However, domain knowledge is not always available in reality. In this work, we propose a general framework for generating an alternative view of the original graph without relying on existing augmentation techniques that may either 1) change the semantics of the original graph or 2) require domain knowledge.

Problem Statement

Notations. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, ..., v_N\}$ represents the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the set of edges. $\mathcal{G}$ is associated with a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, and an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ where $\mathbf{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ otherwise.

Task: Unsupervised Graph Representation Learning. Given a graph $\mathcal{G}$ along with $\mathbf{X}$ and $\mathbf{A}$, we aim to learn a encoder $f(\cdot)$ that produces node embeddings $\mathbf{H} = f(\mathbf{X}, \mathbf{A}) \in \mathbb{R}^{N \times D}$, where $D << F$. In particular, our goal is to learn node embeddings that generalize well to various downstream tasks without using any class information.

Preliminary: Bootstrap Your Own Latent

Before explaining details of our proposed method, we begin by introducing BYOL (Grill et al. 2020), which is the backbone of our proposed framework. The core idea of BYOL is to learn representations of images without using negative samples (Grill et al. 2020). Given two augmented views of an image, BYOL trains two separate encoders, i.e., online encoder f_θ and target encoder f_ξ, and learns representations of images by maximizing the similarity of the two representations produced by each encoder. More formally, BYOL generates two views $\mathbf{x}_1 \sim t(\mathbf{x})$, and $\mathbf{x}_2 \sim t'(\mathbf{x})$ of an image $\mathbf{x}$ given a set of transformations $t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$, and these two generated views of an image are fed into the online and target encoders. Precisely, the online encoder f_θ produces online representation $\mathbf{h}_1 = f_\theta(\mathbf{x}_1)$, while the target encoder f_ξ produces target representation $\mathbf{h}_2 = f_\xi(\mathbf{x}_2)$. Then, both online and target representations are projected to smaller representations $\mathbf{z}_1 = g_\theta(\mathbf{h}_1)$ and $\mathbf{z}_2 = g_\xi(\mathbf{h}_2)$ using projectors g_θ and g_ξ, respectively. Finally, an additional predictor q_θ is applied on top of the projected online representation, i.e., $\mathbf{z_1}$, to make the architecture asymmetric. The objective function is defined as $\mathcal{L}_{\theta,\xi} = \|\bar{q}_\theta(\mathbf{z_1}) - \bar{\mathbf{z}}_2\|^2$, where $\bar{q}_\theta(\mathbf{z_1})$ and $\bar{\mathbf{z}}_2$ denote $l2$-normalized form of $q_\theta(\mathbf{z_1})$ and $\mathbf{z}_2$, respectively. A symmetric loss $\widetilde{\mathcal{L}}_{\theta,\xi}$ is obtained by feeding $\mathbf{x_2}$ into the online encoder and $\mathbf{x_1}$ into the target encoder, and the final objective is to minimize $\mathcal{L}^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$. At each training iteration, a stochastic optimization step is performed to minimize $\mathcal{L}^{\text{BYOL}}_{\theta,\xi}$ with respect to θ only, while ξ is updated using the exponential moving average (EMA) of θ, which is empirically shown to prevent the collapsing problem (Chen and He 2021). More formally, the parameters of BYOL are updated as $\theta \leftarrow \text{optimizer}\left(\theta, \nabla_\theta \mathcal{L}^{\text{BYOL}}_{\theta,\xi}, \eta\right), \ \xi \leftarrow \tau\xi + (1-\tau)\theta$, where η is learning rate for online network, and $\tau \in [0, 1]$ is the decay rate that controls how close ξ remains to θ.

Proposed Method

We first introduce how BYOL has been previously employed on graphs (Thakoor et al. 2021), and discuss about the several limitations of augmentation-based methods for graphs. Finally, we present our proposed method, called AFGRL.

Generating Alternative Views via Augmentation. BGRL (Thakoor et al. 2021) is a recently proposed fully non-contrastive method for learning node representations that does not leverage negative samples benefiting from the framework of BYOL (Grill et al. 2020). Precisely, BGRL generates two different views of a graph via manual augmentations, i.e., node feature masking and edge masking as done by previous methods (Zhu et al. 2020, 2021), and the amount by which the graphs are augmented is controlled by a set of hyperparameters. Then, two encoders, i.e., online and target encoders, generate embeddings given the augmented views of a graph as inputs, and the two generated embeddings are learned to be close to each other. To prevent the representations from collapsing to trivial solutions, BGRL introduces a symmetry-breaking technique (refer to Section for more detailed explanation). It is also worth noting that BGRL intentionally considers simple augmentation techniques to validate the benefit of fully non-contrastive scheme applied on graphs.

Limitation of Augmentation-based Methods on Graphs. Although BGRL has been shown to be effective in a fully non-contrastive manner, i.e., without using negative samples, we observe that *the quality of the learned node representations relies on the choice of the augmentation scheme*. In other words, performance on various downstream tasks evaluated based on the representations learned by BGRL varies greatly according to the choice of hyperparameters associated with augmentations, and the best hyperparameters are different for different datasets. This phenomenon

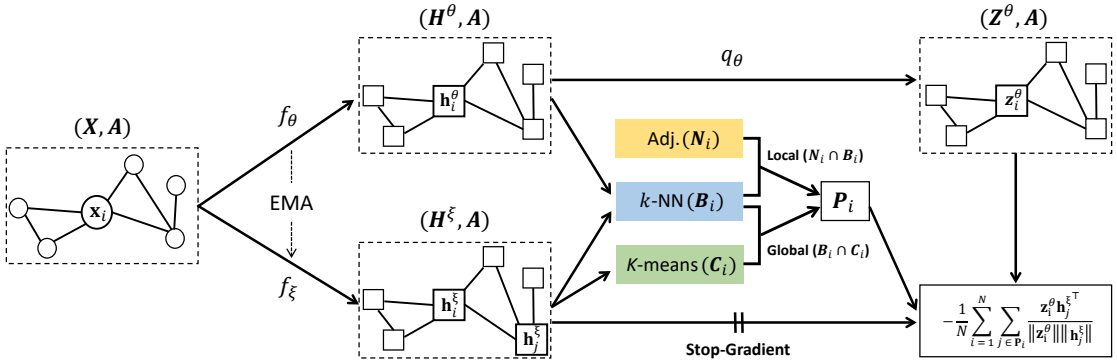


Figure 2: The overall architecture of AFGRL. Given a graph, f_θ and f_ξ generate node embeddings $\mathbf{H}^\theta$ and $\mathbf{H}^\xi$ both of which are used to obtain k-NNs for node v_i, i.e., $\mathbf{B}_i$. Combining it with $\mathbf{N}_i$, we obtain local positives, i.e., $\mathbf{B}_i \cap \mathbf{N}_i$. To obtain global positives for node v_i, K-means clustering is performed on $\mathbf{H}^\xi$, and the result $\mathbf{C}_i$ is combined with $\mathbf{B}_i$, i.e., $\mathbf{B}_i \cap \mathbf{C}_i$. Finally, we combine local and global positives to obtain real positives, i.e., $\mathbf{P}_i$. A predictor q_θ projects $\mathbf{H}^\theta$ to $\mathbf{Z}^\theta$, which is used to compute the final loss along with $\mathbf{H}^\xi$. Note that f_θ is updated via gradient descent of the loss, whereas f_ξ is updated via EMA of f_θ.

		Comp.	Photo	CS	Physics
Node Classi.	BGRL	-4.00%	-1.06%	-0.20%	-0.69%
	GCA	-19.18%	-5.48%	-0.27%	OOM
Node Clust.	BGRL	-11.57%	-13.30%	-0.78%	-6.46%
	GCA	-26.28%	-23.27%	-1.64%	OOM

Table 1: Performance sensitivity according to the hyperparameters for augmentations (i.e., edge drop and node feature masking) on node classification and clustering. Each value indicates the relative performance difference between the best vs. worst performing cases, i.e. $-\frac{(\text{best}-\text{worst})}{\text{best}} \times 100$. The hyperparameters (i.e., probability of edge drop and node feature masking) are chosen within the range from 0.0 to 0.5 to prevent a significant distortion of graphs.

becomes even clearer when stronger augmentations, such as diffusion (Hassani and Khasahmadi 2020), adaptive techniques (Zhu et al. 2021) and the infusion of domain knowledge (Sun et al. 2021a) are applied. Table 1 shows how the performance of augmentation-based methods varies according to the hyperparameters associated with augmentations. More precisely, we report the relative performance of the best performing case compared to the worst performing case, i.e., -4.00% indicates that the worst case performs 4% worse than the best case. We observe that the performance in both tasks is sensitive to the hyperparameters, and that it aggravates when a stronger augmentation technique is employed, i.e., GCA, in which case the role of augmentation becomes even more important. Thus, we need a more stable and general framework for generating an alternative view of the original graph without relying on augmentation techniques introduced in existing works.

Augmentation-Free GRL (AFGRL)

We propose a simple yet effective self-supervised learning framework for generating an alternative view of the original graph taking into account the relational inductive bias of graph-structured data, and the global semantics of graphs. For each node $v_i \in \mathcal{V}$ in graph $\mathcal{G}$, we discover nodes that can serve as positive samples based on the node representations learned by the two encoders. i.e., online encoder $f_\theta(\cdot)$ and target encoder $f_\xi(\cdot)$[1]. More precisely, these encoders initially receive the adjacency matrix $\mathbf{A}$ and the feature matrix $\mathbf{X}$ of the original graph as inputs, and compute the online and target representations. i.e., $\mathbf{H}^\theta = f_\theta(\mathbf{X}, \mathbf{A})$ and $\mathbf{H}^\xi = f_\xi(\mathbf{X}, \mathbf{A})$ whose i-th rows, $\mathbf{h}_i^\theta$ and $\mathbf{h}_i^\xi$, are representations for node $v_i \in \mathcal{V}$. Then, for a given query node $v_i \in \mathcal{V}$, we compute the cosine similarity between all other nodes in the graph as follows:

$$sim(v_i, v_j) = \frac{\mathbf{h}_i^\theta \cdot \mathbf{h}_j^\xi}{\|\mathbf{h}_i^\theta\| \|\mathbf{h}_j^\xi\|}, \forall v_j \in \mathcal{V} \tag{1}$$

where the similarity is computed between the online and the target representations. Given the similarity information, we search for k-nearest-neighbors for each node v_i, and denote them by a set $\mathbf{B}_i$, which can serve as positive samples for node v_i. Essentially, we expect the nearest neighbors in the representation space to belong to the same semantic class as the query node, i.e., v_i in this case. Although $\mathbf{B}_i$ can serve as a reasonable set of positive candidates for node v_i, 1) *it is inherently noisy* as we do not leverage any label information, i.e., $\mathbf{B}_i$ contains samples that are not semantically related to the query node v_i. Moreover, only resorting to the nearest neighbors in the representation space may not only overlook 2) *the structural information inherent in the graph, i.e., relational inductive bias*, but also 3) *the global semantics of the graph*. To address these limitations, we introduce a mechanism to filter out false positives from the samples discovered by k-NN search, while also capturing the local structural information and the global semantics of graphs.

[1] AFGRL adopts the architecture of BGRL (Thakoor et al. 2021), which is slightly modified from BYOL (Grill et al. 2020). In particular, projection networks, i.e., $g_\theta(\cdot)$ and $g_\xi(\cdot)$ are not used.

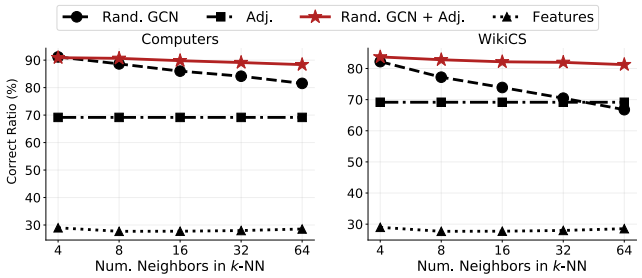


Figure 3: Analysis on the ratio of its neighboring nodes being the same label as the query node across different ks.

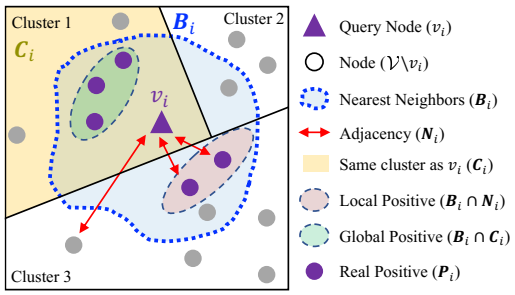


Figure 4: An overview of obtaining real positives of node v_i.

Capturing Local Structural Information. Recall that we expected the nearest neighbors found by k-NN search, i.e., $\mathbf{B}_i$, to share the same class label as the query node v_i. To verify whether our expectation holds, we perform analysis on two datasets, i.e., Amazon Computers and WikiCS datasets as shown in Figure 3. First, we obtain node embeddings from a randomly initialized 2-layer GCN (Kipf and Welling 2016), i.e., $\mathbf{H}_{\text{Rand-GCN}} = \text{Rand-GCN}(\mathbf{X}, \mathbf{A})$, and perform k-NN search for each node given the node embeddings $\mathbf{H}_{\text{Rand-GCN}}$. Then, for each node, we compute the ratio of its neighboring nodes being the same label as the query node. In Figure 3, we observe that although the ratio is high when considering only a small number of neighbors, e.g., $k = 4$, the ratio decreases as k gets larger in both datasets. This implies that although our expectation holds to some extent, there still exists noise.

Hence, to filter out false positives from the nearest neighbors found by k-NN search, i.e., $\mathbf{B}_i$ for each node v_i, we leverage the local structural information among nodes given in the form of an adjacency matrix. i.e., relational inductive bias. More precisely, for a node v_i, its adjacent nodes $\mathbf{N}_i$ tend to share the same label as the query node v_i, i.e., smoothness assumption (Zhu, Ghahramani, and Lafferty 2003). In Figure 3, we indeed observe that the ratio of the adjacent nodes being the same label as the query node ("*Adj*") is about 70% in both datasets, which demonstrates the validity of the smoothness assumption. Therefore, to capture the relational inductive bias reflected in the smoothness assumption, while filtering out false positives from noisy nearest neighbors, we compute the intersection between the nearest neighbors and adjacent nodes, i.e., $\mathbf{B}_i \cap \mathbf{N}_i$. We denote the set of these intersecting nodes as ***local positives*** of v_i. Indeed, Figure 3 shows that the local positives ("*Rand. GCN + Adj*") consistently maintain high correct ratio even when k increases.

Capturing Global Semantics. To capture the semantics of nodes in a global perspective, we leverage clustering techniques. The intuition is to discover non-adjacent nodes that share the global semantic information with the query node. For example, in an academic collaboration network whose nodes denote authors and edges denote collaboration between authors, even though two authors work on the same research topic (i.e., same label), they may not be connected in the graph since they neither collaborated in the past nor share any collaborators. We argue that such semantically similar entities that do not share an edge can be discovered via clustering in a global perspective. In this regard, we apply K-means clustering algorithm on the target representation $\mathbf{H}^{\xi}$ to cluster nodes into a set of K clusters, i.e. $\mathbf{G} = \{G_1, G_2, ..., G_K\}$, and $c(\mathbf{h}_i^{\xi}) \in \{1, ..., K\}$ denotes the cluster assignment of $\mathbf{h}_i^{\xi}$, i.e., $v_i \in G_{c(\mathbf{h}_i^{\xi})}$. Then, we consider the set of nodes that belong to the same cluster as v_i, i.e., $\mathbf{C}_i = \{v_j | v_j \in G_{c(\mathbf{h}_i^{\xi})}\}$, as its semantically similar nodes in the global perspective. Finally, we obtain the intersection between the nearest neighbors and the semantically similar nodes in the global perspective, i.e., $\mathbf{B}_i \cap \mathbf{C}_i$, and we denote the set of these intersecting nodes as ***global positives*** of v_i. In other words, nodes that are among the nearest neighbors of v_i and at the same time belong to the same cluster as v_i are considered as globally positive neighbors. It is important to note that as K-means clustering algorithm is sensitive to the cluster centroid initialization, we perform multiple runs to ensure robustness of the clustering results. Specifically, we perform K-means clustering M times and obtain M sets of clusters, i.e., $\{\mathbf{G}^{(j)}\}_{j=1}^{M}$, where $\mathbf{G}^{(j)} = \left\{G_1^{(j)}, G_2^{(j)}, ..., G_K^{(j)}\right\}$ is the result of j-th run of the clustering. Then, we define $\mathbf{C}_i = \bigcup_{j=1}^{M} G^{(j)}_{c^{(j)}(\mathbf{h}_i^{\xi})}$, where $c^{(j)}(\mathbf{h}_i^{\xi}) \in \{1, ..., K\}$ denotes the cluster assignment of $\mathbf{h}_i^{\xi}$ in the j-th run of clustering.

Objective Function. In order to consider both the local and global information, we define the set of ***real positives*** for node v_i as follows:

$$\mathbf{P}_i = (\mathbf{B}_i \cap \mathbf{N}_i) \cup (\mathbf{B}_i \cap \mathbf{C}_i) \tag{2}$$

Our objective function aims to minimize the cosine distance between the query node v_i and its real positives $\mathbf{P}_i$:

$$\mathcal{L}_{\theta,\xi} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{v_j \in \mathbf{P}_i} \frac{\mathbf{z}_i^{\theta}\mathbf{h}_j^{\xi\top}}{\left\|\mathbf{z}_i^{\theta}\right\| \left\|\mathbf{h}_j^{\xi}\right\|}, \tag{3}$$

where $\mathbf{z}_i^{\theta} = q_{\theta}(\mathbf{h}_i^{\theta}) \in \mathbb{R}^D$ is the prediction of the online embedding $\mathbf{h}_i^{\theta} \in \mathbb{R}^D$, and $q_{\theta}(\cdot)$ is the predictor network. Following BYOL, AFGRL's online network is updated based on the gradient of its parameters with respect to the loss function (Equation 3), while the target network

	WikiCS	Computers	Photo	Co.CS	Co.Physics
Sup. GCN	77.19 ± 0.12	86.51 ± 0.54	92.42 ± 0.22	93.03 ± 0.31	95.65 ± 0.16
Raw feats.	71.98 ± 0.00	73.81 ± 0.00	78.53 ± 0.00	90.37 ± 0.00	93.58 ± 0.00
node2vec	71.79 ± 0.05	84.39 ± 0.08	89.67 ± 0.12	85.08 ± 0.03	91.19 ± 0.04
DeepWalk	74.35 ± 0.06	85.68 ± 0.06	89.44 ± 0.11	84.61 ± 0.22	91.77 ± 0.15
DW + feats.	77.21 ± 0.03	86.28 ± 0.07	90.05 ± 0.08	87.70 ± 0.04	94.90 ± 0.09
DGI	75.35 ± 0.14	83.95 ± 0.47	91.61 ± 0.22	92.15 ± 0.63	94.51 ± 0.52
GMI	74.85 ± 0.08	82.21 ± 0.31	90.68 ± 0.17	OOM	OOM
MVGRL	77.52 ± 0.08	87.52 ± 0.11	91.74 ± 0.07	92.11 ± 0.12	95.33 ± 0.03
GRACE	**77.97 ± 0.63**	86.50 ± 0.33	92.46 ± 0.18	92.17 ± 0.04	OOM
GCA	77.94 ± 0.67	87.32 ± 0.50	92.39 ± 0.33	92.84 ± 0.15	OOM
BGRL	76.86 ± 0.74	89.69 ± 0.37	93.07 ± 0.38	92.59 ± 0.14	95.48 ± 0.08
AFGRL	77.62 ± 0.49	**89.88 ± 0.33**	**93.22 ± 0.28**	**93.27 ± 0.17**	**95.69 ± 0.10**

Table 2: Performance on node classification (OOM: Out of memory on 24GB RTX3090).

is updated by smoothing the online network. We also symmetrize the loss function. In the end, the online embeddings, i.e., $\mathbf{H}^{\theta} \in \mathbb{R}^{N \times D}$ are used for downstream tasks. Figure 4 illustrates the overview of obtaining real positives of node v_i.

In summary, 1) AFGRL does not rely on arbitrary augmentation techniques for the model training, thereby achieving stable performance. 2) AFGRL filters out false positives from the samples discovered by k-NN search, while also capturing the local structural information, i.e., relational inductive bias, and the global semantics of graphs. 3) AFGRL does not require negative samples for the model training, thereby avoiding sampling bias and alleviating computational/memory costs suffered by previous contrastive methods.

		GRACE	GCA	BGRL	AFGRL
WikiCS	NMI	**0.4282**	0.3373	0.3969	0.4132
	Hom.	**0.4423**	0.3525	0.4156	0.4307
Computers	NMI	0.4793	0.5278	0.5364	**0.5520**
	Hom.	0.5222	0.5816	0.5869	**0.6040**
Photo	NMI	0.6513	0.6443	**0.6841**	0.6563
	Hom.	0.6657	0.6575	**0.7004**	0.6743
Co.CS	NMI	0.7562	0.7620	0.7732	**0.7859**
	Hom.	0.7909	0.7965	0.8041	**0.8161**
Co.Physics	NMI	OOM	OOM	0.5568	**0.7289**
	Hom.	OOM	OOM	0.6018	**0.7354**

Table 3: Performance on node clustering in terms of NMI and homogeneity.

Experiments

Experimental Setup

Datasets. To evaluate AFGRL, we conduct experiments on five widely used datasets, including WikiCS (Mernyei and Cangea 2020), Amazon (*Computers* and *Photo*) (McAuley et al. 2015), Coauthor (*Co.CS* and *Co.Physics*) (Sinha et al. 2015).

Methods Compared. We primarily compare AFGRL against GRACE (Zhu et al. 2020), BGRL (Thakoor et al. 2021) and GCA (Zhu et al. 2021), which are the current state-of-the-art self-supervised methods for learning representations of nodes in a graph. For all baselines but BGRL, we use the official code published by the authors. As the official code for BGRL is not available, we implement it by ourselves, and try our best to reflect the details provided in the original paper (Thakoor et al. 2021). We also report previously published results of other representative methods, such as DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), DGI (Veličković et al. 2018), GMI (Peng et al. 2020), and MVGRL (Hassani and Khasahmadi 2020), as done in (Thakoor et al. 2021; Zhu et al. 2021).

Evaluation protocol. We evaluate AFGRL on three node-level tasks, i.e., node classification, node clustering and node similarity search. We first train all models in an unsupervised manner. For node classification, we use the learned embeddings to train and test a simple logistic regression classifier (Veličković et al. 2018). We report the test performance when the performance on validation data gives the best result. For node clustering and similarity search, we perform evaluations on the learned embeddings at every epoch and report the best performance.

Implementation details. We use a GCN (Kipf and Welling 2016) model as the encoders, i.e., $f_\theta(\cdot)$ and $f_\xi(\cdot)$. More formally, the encoder architecture is defined as:

$$\mathbf{H}^{(l)} = \mathrm{GCN}^{(l)}(\mathbf{X}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{W}^{(l)}), \quad (4)$$

where $\mathbf{H}^{(l)}$ is the node embedding matrix of the l-th layer for $l \in [1, ..., L]$, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\hat{\mathbf{D}} = \sum_i \hat{\mathbf{A}}_i$ is the degree matrix, $\sigma(\cdot)$ is a nonlinear activation function such as ReLU, and $\mathbf{W}^{(l)}$ is the trainable weight matrix for the l-th layer. We perform grid-search on several hyperparameters, such as learning rate η, decay rate τ, node embedding dimension size D, number of layers of GCN encoder L, for fair comparisons.

		GRACE	GCA	BGRL	AFGRL
WikiCS	Sim@5	0.7754	0.7786	0.7739	**0.7811**
	Sim@10	0.7645	**0.7673**	0.7617	0.7660
Computers	Sim@5	0.8738	0.8826	0.8947	**0.8966**
	Sim@10	0.8643	0.8742	0.8855	**0.8890**
Photo	Sim@5	0.9155	0.9112	**0.9245**	0.9236
	Sim@10	0.9106	0.9052	**0.9195**	0.9173
Co.CS	Sim@5	0.9104	0.9126	0.9112	**0.9180**
	Sim@10	0.9059	0.9100	0.9086	**0.9142**
Co.Physics	Sim@5	OOM	OOM	0.9504	**0.9525**
	Sim@10	OOM	OOM	0.9464	**0.9486**

Table 4: Performance on similarity search. (Sim@n: Average ratio among n nearest neighbors sharing the same label as the query node.)

Performance Analysis

Overall evaluation. Table 2 shows the node classification performance of various methods. We have the following observations: **1)** Our augmentation-free AFGRL generally performs well on all datasets compared with augmentation-based methods, i.e., GRACE, GCA and BGRL, whose reported results are obtained by carefully tuning the augmentation hyperparameters. Recall that in Table 1 we demonstrated that their performance is highly sensitive to the choice of augmentation hyperparameters. This verifies the benefit of our augmentation-free approach. **2)** We also evaluate AFGRL on node clustering (Table 3) and similarity search (Table 4). Note that the best hyperparameters for node classification task were adopted. Table 3 shows that AFGRL generally outperforms other methods in node clustering task. We argue that this is mainly because AFGRL also considers global semantic information unlike the compared methods. **3)** It is worth noting that methods built upon instance discrimination principle (Wu et al. 2018), i.e., GRACE and GCA, are not only memory consuming (OOM on large datasets), but also generally perform worse than their counterparts on various tasks (especially on clustering). This indicates that instance discrimination, which treats all other nodes except itself as negatives without considering the graph structural information, is not appropriate for graph-structured data, especially for clustering in which the global structural information is crucial. **4)** AFGRL generally performs well on node similarity search (Table 4). This is expected because AFGRL aims to make nearest neighbors of each node share the same label with the query node by discovering the local and the global positives.

Ablation Studies. To verify the benefit of each component of AFGRL, we conduct ablation studies on two datasets that exhibit distinct characteristics, i.e., Amazon Computers (E-commerce network) and WikiCS (Reference network). In Figure 5, we observe that considering both local structural and global semantic information shows the best performance. Moreover, we observe that the global semantic information is more beneficial than the local structural information. This can be explained by the performance of "*k-NN only*" variant, which performs on par with "*k-NN + Adj*" variant. That is, we conjecture that performing k-NN on the node representations learned by our framework can capture sufficient local structural information contained in the adjacency matrix. Based on the ablation studies, we argue that AFGRL still gives competitive performance even when the adjacency matrix is sparse, which shows the practicality of our proposed framework. Finally, the low performance of "*Clus-only*" variant implies the importance of considering the local structural information in graph-structured data.

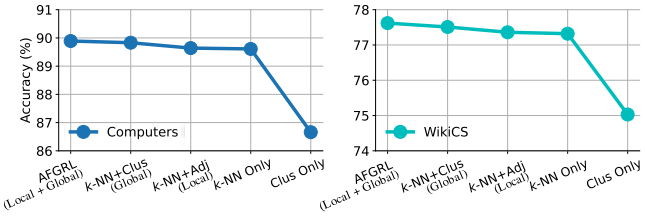


Figure 5: Ablation study on AFGRL.

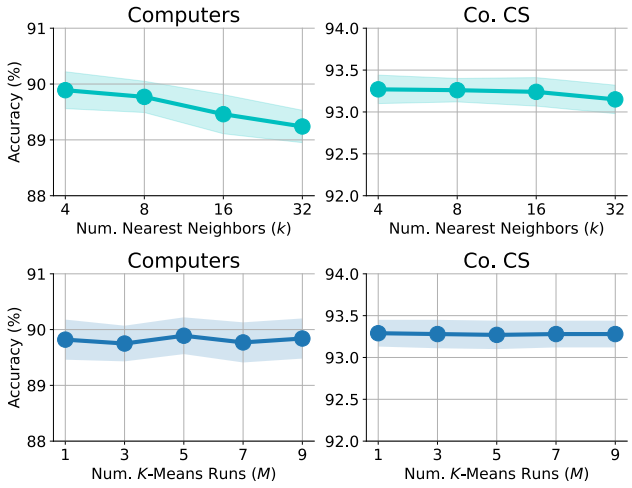


Figure 6: Sensitivity analysis.

Hyperparameter Analysis. Figure 6 shows the sensitivity analysis on the hyperparameters k and M of AFGRL. We observe that $k = 4$ and $M > 1$ generally give the best performance, while the performance is rather stable over various Ms. This verifies that our augmentation-free approach can be easily trained compared with other augmentation-

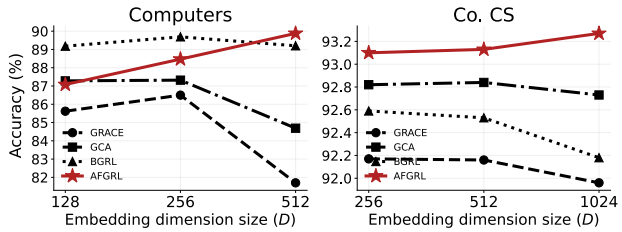
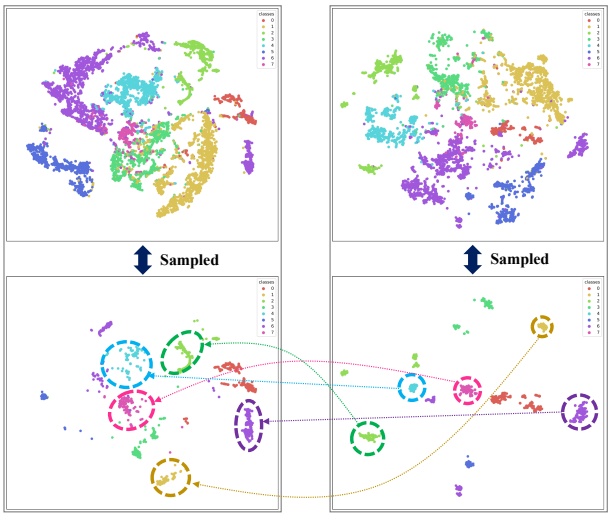


Figure 7: Effect of embedding dimension size (D).



(a) GCA (b) AFGRL

Figure 8: t-SNE embeddings of nodes in *Photo* dataset.

based methods, i.e., stable over hyperparameters, while outperforming them in most cases. Moreover, in Figure 7, we conduct experiments across various sizes of node embedding dimension D. We observe that AFGRL benefits from high-dimensional embeddings, while other methods rapidly saturate when the dimension of embeddings increase. Note that Zbontar et al. (2021) recently showed similar results indicating that methods based on instance discrimination (Wu et al. 2018) is prone to the curse of dimensionality.

Visualization of embeddings. To provide a more intuitive understanding of the learned node embeddings, we visualize node embeddings of GCA (Figure 8(a)) and AFGRL (Figure 8(b)) by using t-SNE (Van der Maaten and Hinton 2008). Each point represents a node, and the color represents the node label. We observe that node embeddings generated by both methods are grouped together according to their corresponding node labels. However, the major difference is that AFGRL captures more fine-grained class information compared with GCA. That is, for AFGRL, there tend to be small clusters within each label group. To emphasize this, we sample the same set of nodes from each label, and compare their embeddings (Figure 8 bottom). We clearly observe that nodes are more tightly grouped in AFGRL compared with GCA, which implies that AFGRL captures more fine-grained class information.

Conclusion

In this paper, we propose a self-supervised learning framework for graphs, which requires neither augmentation techniques nor negative samples for learning representations of graphs. Instead of creating two arbitrarily augmented views of a graph and expecting them to preserve the semantics of the original graph, AFGRL discovers nodes that can serve as positive samples by considering the local structural information and the global semantics of graphs. The major benefit of AFGRL over other self-supervised methods on graphs is its stability over hyperparameters while maintaining competitive performance even without using negative samples for the model training, which makes AFGRL practical. Through experiments on multiple graphs on various downstream tasks, we empirically show that AFGRL is superior to the state-of-the-art methods that are sensitive to augmentation hyperparameters.

Acknowledgements

This work was supported by the NRF grant funded by the MSIT (No.2021R1C1C1009081), and the IITP grant funded by the MSIT (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)).

References

Banville, H.; Albuquerque, I.; Hyvärinen, A.; Moffat, G.; Engemann, D.-A.; and Gramfort, A. 2019. Self-supervised representation learning from electroencephalography signals. In *MLSP*, 1–6. IEEE.

Banville, H.; Chehab, O.; Hyvärinen, A.; Engemann, D.-A.; and Gramfort, A. 2021. Uncovering the structure of clinical EEG signals with self-supervised learning. *Journal of Neural Engineering*, 18(4): 046020.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Bielak, P.; Kajdanowicz, T.; and Chawla, N. V. 2021. Graph Barlow Twins: A self-supervised representation learning framework for graphs. *arXiv preprint arXiv:2106.02466*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

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

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 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*.

Jiao, Y.; Xiong, Y.; Zhang, J.; Zhang, Y.; Zhang, T.; and Zhu, Y. 2020. Sub-graph contrast for scalable self-supervised graph representation learning. In *ICDM*, 222–231. IEEE.

Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; and Pan, S. 2021. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. *arXiv preprint arXiv:2105.05682*.

Jing, B.; Park, C.; and Tong, H. 2021. Hdmi: High-order deep multiplex infomax. In *WWW*, 2414–2424.

Kefato, Z. T.; and Girdzijauskas, S. 2021. Self-supervised Graph Neural Networks without explicit negative sampling. *arXiv preprint arXiv:2103.14958*.

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

Lin, S.; Zhou, P.; Hu, Z.-Y.; Wang, S.; Zhao, R.; Zheng, Y.; Lin, L.; Xing, E.; and Liang, X. 2021. Prototypical Graph Contrastive Learning. *arXiv preprint arXiv:2106.09645*.

Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, 69–84. Springer.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*, 259–270.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*, 701–710.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, 1150–1160.

Saunshi, N.; Plevrakis, O.; Arora, S.; Khodak, M.; and Khandeparkar, H. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, 5628–5637. PMLR.

Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B.-J.; and Wang, K. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*, 243–246.

Sun, F.-Y.; Hoffmann, J.; Verma, V.; and Tang, J. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*.

Sun, M.; Xing, J.; Wang, H.; Chen, B.; and Zhou, J. 2021a. MoCL: Contrastive Learning on Molecular Graphs with Multi-level Domain Knowledge. *arXiv preprint arXiv:2106.04509*.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Ning, Y.; Yu, P. S.; and He, L. 2021b. SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *WWW*, 2081–2091.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial Graph Augmentation to Improve Graph Contrastive Learning. *arXiv preprint arXiv:2106.05819*.

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*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

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*.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Xie, Y.; Xu, Z.; Zhang, J.; Wang, Z.; and Ji, S. 2021. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph Contrastive Learning Automated. *arXiv preprint arXiv:2106.07594*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *NeurIPS*, 33: 5812–5823.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*.

Zhao, T.; Liu, Y.; Neves, L.; Woodford, O.; Jiang, M.; and Shah, N. 2020. Data augmentation for graph neural networks. *arXiv preprint arXiv:2006.06830*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 912–919.

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

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *WWW*, 2069–2080.