Multi-Task Curriculum Graph Contrastive Learning with Clustering Entropy Guidance

Chusheng Zeng[1] , **Bocheng Wang**[1] , **Jinghui Yuan**[1] , **Mulin Chen**[1*] and **Xuelong Li**[2]
[1]School of Artifcial Intelligence, OPtics and ElectroNics (iOPEN),
Northwestern Polytechnical University, China
[2]Institute of Artificial Intelligence (TeleAI), China Telecom, China
{zcs, wobocheng, yuanjh}@mail.nwpu.edu.cn, chenmulin@nwpu.edu.cn, xuelong_li@ieee.org

Abstract

Recent advances in unsupervised deep graph clustering have been significantly promoted by contrastive learning. Despite the strides, most graph contrastive learning models face challenges: 1) graph augmentation is used to improve learning diversity, but commonly used random augmentation methods may destroy inherent semantics and cause noise; 2) the fixed positive and negative sample selection strategy ignores the difficulty distribution of samples when deal with complex real data, thereby impeding the model's capability to capture fine-grained cluster patterns. To reduce these problems, we propose the clustering-guided Curriculum Graph contrastive Learning (CurGL) framework. CurGL uses clustering entropy as the guidance of the following graph augmentation and contrastive learning. Specifically, according to the clustering entropy, the intra-class edges and important features are emphasized in augmentation. Then, a multi-task curriculum learning scheme is proposed, which employs the clustering guidance to shift the focus from the discrimination task to the clustering task. In this way, the sample selection strategy of contrastive learning can be adjusted adaptively from early to late stage, which enhances the model's flexibility for complex data structure. Experimental results demonstrate that CurGL has achieved excellent performance compared to state-of-the-art competitors.

1 Introduction

In recent years, the representational capacity of graph data has made it prevalent in various applications, including social networks [Majeed and Rauf, 2020; Newman *et al.*, 2002], knowledge graphs [Ji *et al.*, 2021], and traffic prediction [Zhao *et al.*, 2019; Li *et al.*, 2023]. The prevalence has been further amplified by the emergence of deep graph neural networks [Kipf and Welling, 2016a; Veličković *et al.*, 2017], which has enabled the efficient analysis of complex graph data. Specifically, unsupervised deep graph learning [Wang *et al.*, 2021b; Liu *et al.*, 2022a; Mo *et al.*, 2022; Chen and Li, 2022] has garnered extensive research interest due to its ability to extract discriminative, and interpretable graph features.

Graph contrastive learning is a fundamental paradigm within the field of unsupervised graph learning, and consists of two parts including graph augmentation and contrastive learning. Data augmentation changes the original data graph to obtain multiple views with similar semantics, thereby expanding the selection space of positive and negative sample pairs [Xu *et al.*, 2024; Wang *et al.*, 2021c]. For example, many data augmentation methods select the nodes and edges randomly, and delete/mask/disturb them to produce an augmented graph [Zhu *et al.*, 2020; You *et al.*, 2020]. Contrastive learning enhances the similarity of samples with related semantics, while pushing samples with low relevance away from each other. The above process helps to learn graph embeddings with high discriminability, which in turn reveals the implicit structure of the data. The early methods [Hjelm *et al.*, 2019; Veličković *et al.*, 2018] take the non-corresponding nodes among views as negatives, which can mistakenly pull many intra-class samples far apart, resulting in sampling bias. Some recent methods [Lin *et al.*, 2022; Zhao *et al.*, 2021] have attempted to improve the selection strategies for positive and negative samples, adopting specific strategies to select more appropriate positive and negative samples, such as taking the intra-class nodes as positives and the inter-class nodes as negatives.

Existing methods for graph contrastive learning often encounter difficulties when dealing with complex data. On the one hand, the widely used random graph data augmentation methods [Fang *et al.*, 2023] can potentially remove intra-class edges and mask features essential for representing the class. Unlike ideal augmentation, this approach can disrupt the graph structure critical to clustering, thereby limiting the performance of contrastive learning frameworks. On the other hand, most methods adopt the fixed strategy to select positive and negative samples throughout the training process, which is not flexible in practical applications. For example, two common strategies for positive/negative samples selection in contrastive learning are: using corresponding samples from other views as positives and treating all others as negatives [Zhu *et al.*, 2020; You *et al.*, 2020; **?**], or using pseudo-labels generated during training to de-

*Corresponding author.

fine positive and negative samples [Yang *et al.*, 2023; Lin *et al.*, 2022; Xia *et al.*, 2022]. The first approach focuses on learning the discriminative features for each node. However, taking all non-corresponding samples as negatives hinders intra-class compactness. The second approach employs pseudo-labels generated by simple clustering methods to select positive/negative samples, which may be unreliable in the early stages due to the insufficient discriminability of the learned embeddings. The limitation of fixed strategy hinders the model's ability to explore the intrinsic representation and complex topological relationship, potentially leading the model to converge to a suboptimal solution for clustering.

To alleviate the above problems, we propose clustering-guided Curriculum Graph contrastive Learning (CurGL) framework. As shown in Fig. 1, the intermediate results of the model are clustered to calculate the clustering entropy, which serves as a clustering guidance for the whole framework. On this basis, a clustering-friendly strategy is adopted for performing structure-level and feature-level data augmentation. Then, the multi-task curriculum learning scheme uses clustering entropy to determine the clustering confidence of samples and place them in different contrastive learning tasks. As training progresses, samples are transited from the early stage simple discrimination task to the more challenging clustering task in the late stages, such that the flexible adjustment of sample selection is achieved. The main contributions of this paper are as follows.

- A clustering-guided graph contrastive learning framework is established. The clustering entropy is defined to serve as the clustering guidance, which is used to evaluate the importance and clustering confidence of nodes throughout the whole framework.
- A clustering-friendly graph augmentation method is proposed. Under the clustering guidance, structure augmentation tends to preserve intra-class edges, while feature augmentation is more likely to retain the class-specific features.
- A multi-task curriculum learning scheme is developed to explore the complex data structure. It allows the model to first learn discriminative representations of the samples and then move towards clustering optimization. The dynamic contrastive learning effectively enhances the capability to capture cluster-oriented discriminative features.

2 Related Work

2.1 Graph Contrastive Learning

After making progress in image recognition[Zbontar *et al.*, 2021; Wang and Qi, 2022], contrastive learning combined with Graph Neural Networks (GNNs) has also garnered significant attention from many researchers. Explorations in graph contrastive learning have primarily concentrated on graph augmentation technologies and contrastive objectives.

Data augmentation enriches the diversity of training samples by generating different views. MVGRL [Hassani and Khasahmadi, 2020] and DCRN [Liu *et al.*, 2022b] utilize graph diffusion to create augmented views, while methods like GRACE [Zhu *et al.*, 2020] and SCAGC [Xia *et al.*, 2022] achieve this through random attribute and edge perturbations. Most augmentation techniques are often stochastic and uncontrollable, potentially disrupting semantics and cause noise. Recently, several adaptive augmentation methods [Zhang *et al.*, 2024] have been proposed. GCA [Zhu *et al.*, 2021] learns the weights of discarding adaptively, with the expectation that the model's final learned representations will be insensitive to unimportant nodes or edges. However, It is not necessarily suitable for downstream clustering tasks. CurGL is designed with clustering as the objective, aiming to obtain augmented views that better align with the underlying clustering structure.

Contrastive objective learns the embedding by constructing pairs of positive and negative samples. MVGRL designs an InfoMax loss to maximize the cross-view mutual information between nodes and the global summary of the graph. AGE[Cui *et al.*, 2020] devises a pretext task, using a cross-entropy loss to classify similar and dissimilar nodes. Based on k-Means or other graph-based clustering methods[Wang *et al.*, 2021a], some methods[Lin *et al.*, 2022; Yang *et al.*, 2023] use intermediate clustering results to guide the contrast objective. CurGL designed two types of contrastive objectives to tackle tasks of varying difficulties, and we employ curriculum learning to tailor the tasks for each node based on the model's current learning state.

2.2 Curriculum Learning

Curriculum learning [Bengio *et al.*, 2009; Wang *et al.*, 2021d] is a training strategy that mimics the human learning process. It allows the model to start with simpler samples and gradually progress to more complex ones, as well as to advanced knowledge. Some studies [Jiang *et al.*, 2018; Han *et al.*, 2018] have theoretically demonstrated the effectiveness of curriculum learning in enhancing generalization capabilities when dealing with noisy data. Curriculum learning is widely applied across various areas of machine learning[Zbontar *et al.*, 2021] and deep learning[Matiisen *et al.*, 2019; Graves *et al.*, 2017].

Curriculum Learning consists of two main parts: difficulty measurer and training scheduler[Hacohen and Weinshall, 2019]. The difficulty measurer is used to evaluate the difficulty of the sample. Predefined difficulty measurer [Platanios *et al.*, 2019; Spitkovsky *et al.*, 2010; Tay *et al.*, 2019] is mainly designed manually according to the data characteristics of a specific task. The training scheduler determines the appropriate training data to feed into the model based on the evaluation results of the difficulty measurer. But existing predefined training schedulers[Cirik *et al.*, 2016] are usually data and task independent, and most curriculum learning in various scenarios uses a similar training scheduler. The discrete scheduler adjusts the training data at each fixed round or when the current data converges, while the continuous scheduler adjusts the training data at each round according to a defined scheduling function. Different from the general curriculum learning that only controls the number of sample participation, CurGL lets samples perform tasks of different difficulty at different stages of the model, which considers the difficulty distribution of samples.

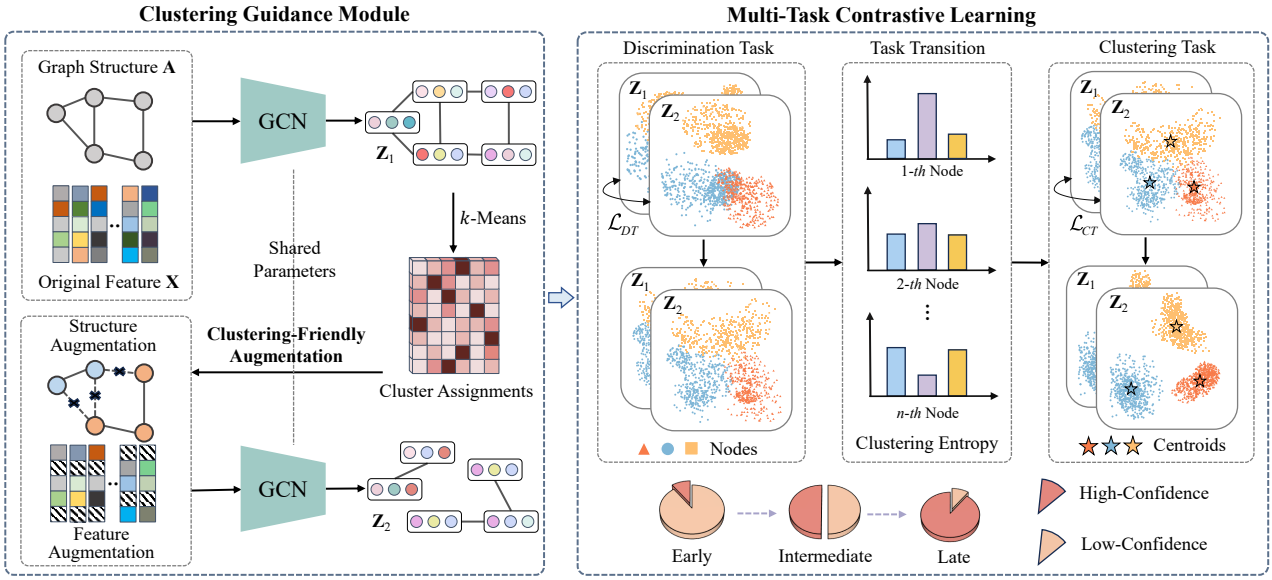


Figure 1: Pipeline of CurGL. The Clustering Guidance Module clusters the embedding $\mathbf{Z_1}$ to obtain clustering guidance. Clustering-Friendly Augmentation applies clustering-oriented structure augmentation and feature augmentation to the original data. According to the clustering guidance, Crriculum Learning divides the nodes into high confidence groups and low confidence groups to perform different contrastive tasks in Multi-Task Contrastive Learning.

3 Methodology

In this section, we will provide a detailed introduction to the proposed CurGL method. The overall pipeline of CurGL is shown in Fig. 1.

3.1 Notations and Problem Definition

In this paper, uppercase letters represent matrices and lowercase letters represent vectors. An undirected graph with n nodes is defined as $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the original feature matrix of the nodes, $\mathbf{A} \in \mathbb{R}^{n \times n}$ represents the adjacency matrix of the graph data.

This paper aims to address an unsupervised graph clustering problem. Specifically, given a graph $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$, the task is to train a GCN encoder such that $\mathbf{Z} = f(\mathbf{X}, \mathbf{A})$, and the resulting $\mathbf{Z}$ can be used to cluster all nodes $\mathcal{V}$ of the graph $\mathcal{G}$ into k classes.

3.2 Clustering Guidance Module

The clustering guidance module is introduced to make the learned embeddings more suitable for clustering tasks. Clustering entropy is defined to guide subsequent augmentation and contrastive learning. It also serves as a loss function to improve the quality of pseudo-labels.

Given the original graph data $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$ and embedding $\mathbf{Z_1}$ computed by GCN, performing k-Means on $\mathbf{Z_1}$ results in k cluster centroids $\mathbf{C} = \{\{c_1\}, \{c_2\}, \ldots, \{c_k\}\}$ and pseudo-labels L of the samples. Then, we calculate the probability assignment matrix $\mathbf{P}$ by

$$\mathbf{P} = softmax(\mathbf{Z_1} \cdot \mathbf{C^T}), \tag{1}$$

where $\mathbf{Z_1} \cdot \mathbf{C^T}$ calculates the dot product similarity between each sample and the cluster center, the $softmax(\cdot)$ performs exponential normalization of each row of $\mathbf{Z_1} \cdot \mathbf{C^T}$ to obtain the probability assignment matrix $\mathbf{P} \in \mathbb{R}^{n \times k}$. Each row of $\mathbf{P}$ represents the probability that the corresponding node is assigned to all the centroids.

To assess the quality of the clustering, we propose the clustering entropy

$$E_i = -\sum_{j=1}^{k} P_{ij} \log(P_{ij}). \tag{2}$$

E_i indicates the uncertainty of the probability distribution for each node. Nodes with high E_i have probabilities that are relatively close to each cluster centroid, indicating a lower confidence in the clustering. Conversely, nodes with low E_i have a higher confidence in clustering. For better clustering, the samples with high confidence should increase during the training procedure, such that a low clustering entropy can be achieved. Therefore, a clustering entropy loss is designed to optimize the clustering effect, defined as

$$\mathcal{L}_{EN} = \frac{1}{n} \sum_{i=1}^{n} E_i. \tag{3}$$

Optimizing $\mathcal{L}_{EN}$ can drive the learned embedding $\mathbf{Z}$ to reveal a clear cluster structure, and provide more reliable clustering-oriented guidance for the subsequent modules.

3.3 Clustering-Friendly Augmentation

In this part, we design a clustering-friendly augmentation method, which consists of structure augmentation and feature augmentation. They preserve edges between nodes of the intra-class and important features that are beneficial for the clustering task respectively.

Structure Augmentation. Each edge is assigned with an importance weight to determine its probability of being deleted, which can be formulized as

$$u_e^{ij} = E_i E_j + \mu\delta(L_i - L_j, 0), \tag{4}$$

where u_e^{ij} represents the importance measure between nodes i-th and j-th. The indicator function $\delta(L_i - L_j, 0)$ is 0 when $L_i = L_j$ and 1 otherwise. The meaning of u_e^{ij} is that when the nodes at both ends of the edge have low clustering entropy and belong to the same class, the value of u_e^{ij} is relatively smaller, indicating a higher importance of the edge, and thus a lower probability of being removed. After normalization, the probability of each edge being removed is obtained, and the normalization method is given by

$$p_u^e = \min\left(\frac{u_e^{\max} - u_e}{u_e^{\max} - \bar{u}_e} \cdot p_e, p_\tau\right), \tag{5}$$

where $u_e^{\max}$ and $\bar{u}_e$ are the maximum and average values of u_e respectively. p_e is the overall edge deletion probability, and p_τ is the truncation probability, which is used to prevent the deletion probability of certain edges from being too high.

The structure augmentation emphasizes the edges that connect reliable nodes of the same class. As a result, important clustering-relevant semantic information is preserved.

Feature Augmentation. Feature augmentation aims to preserve the representative class-specific features. For one-hot encoded features, 1 indicates the presence of a feature, while 0 indicates its absence, the frequency of each feature appearing in nodes within the same class reflects the importance of that feature. Therefore the feature weight for the j-th class f_j can be defined as

$$f_j = \sum_{l_i = j} z_i. \tag{6}$$

For features that are not one-hot encoded, the distribution of features within the class can also be used to obtain $\mathbf{f}_j$. If the distribution of a feature is more concentrated among the samples within a class, it will have a higher feature weight f_j, which can be determined by statistical measures such as variance.

We can obtain the probability that each feature within each class is masked by normalizing f_j,

$$p_u^{f_j} = \min\left(\frac{f_j^{\max} - f_j}{f_j^{\max} - \bar{f}_j} \cdot p_f, p_\tau\right), \tag{7}$$

where p_f is the overall probability of feature augmentation and p_τ is the truncation probability.

The augmentation approach emphasizes important features within each cluster, which makes it more convenient to judge the subordinate cluster of each sample. Features irrelevant to clustering are more likely to be removed to reduce noise.

3.4 Multi-Task Curriculum Learning

The multi-task curriculum learning scheme is proposed to deal with complex real data. In order to simulate the real-world knowledge learning process, samples should start with simple contrastive task and gradually turn to complex contrastive task. In the early stages of training, the embeddings are not discriminable, and unsuitable for clustering. Therefore, we start from the discrimination task, and gradually transition towards the clustering task.

Self-Paced Curriculum Learning. In each training iteration of the model, we categorize samples into low-confidence and high-confidence groups to perform the discrimination and clustering task, respectively. A self-paced curriculum learning is used to achieve automatic transformation of tasks according to the clustering entropy.

To distinguish the confidence of samples, we define a indicator vector $v \in \{0, 1\}^n$, where $v_i = 0$ denotes that the i-th sample belongs to the low-confidence group. Self-paced curriculum learning assigns the contrastive task to samples by controlling each element in the indicator vector v. Given n samples, we define n_{CT}^t as the number of samples in the model at the t-th epoch iteration that are selected to perform the clustering task. The samples are chosen based on their ranking in clustering entropy, and their corresponding elements in the indicator vector v are set to 1. Moreover, n_{DT} is the number of samples that perform discrimination task, and $n_{CT} + n_{DT} = n$. Curriculum pace ε is defined to control the number of high-confidence samples. With the total number of iterations T, the n_{CT}^{t+1} is computed as

$$n_{CT}^{t+1} = \min(n_{CT}^t + \varepsilon\frac{n}{T}, n). \tag{8}$$

When ε reaches 1, all samples participate in the clustering task, the n_{CT} will not increase any more.

Discrimination Task. The purpose of discrimination task is to distinguish each sample individually and learn clear self-representations. Since the discrimination task does not consider the topological relationships between samples, it is regarded as a relatively simple task. Samples in the low-confidence group do not exhibit well-defined clustering structures, so we assign them to perform the discrimination task.

To push nodes apart in the embedding space, the positive and negative sampling corresponding to the discrimination task is that the positive sample is the augmented view of the sample itself, while the negative samples are all other nodes. For z_i, the discrimination task loss is

$$\ell_{DT(i)} = -\frac{1}{2}\sum_{j=1}^{2}\log\Big(\frac{e^{\mathcal{S}\left(\left(z_i^1, z_i^2\right)/\tau\right)}}{e^{\mathcal{S}\left(\left(z_i^1, z_i^2\right)/\tau\right)} + \sum_{k\neq i} e^{\mathcal{S}\left(\left(z_i^j, z_k^1\right)/\tau\right)} + \sum_{k\neq i} e^{\mathcal{S}\left(\left(z_i^j, z_k^2\right)/\tau\right)}}\Big), \tag{9}$$

where z_i^j denotes the i-th node in the j-th view, τ is the temperature parameter, $\mathcal{S}(\cdot)$ is the similarity calculation function. The overall discrimination task loss can be expressed as

$$\mathcal{L}_{DT} = \frac{1}{n_{DT}}\sum_{i=1}^{n}(1 - v_i)\ell_{DT(i)}. \tag{10}$$

By minimizing $\mathcal{L}_{DT}$, the low-confidence sample gradually captures the key internal features, resulting in a more discriminative embedding. As the discriminability improves, the

topological relationship becomes more clear, and the samples gradually transform into high-confidence ones. Then, they can participate into the clustering task.

Clustering Task. Once the samples in the embedding space exhibit sufficient discriminability, the pseudo-labels and -centroids obtained by k-Means become more reliable. Therefore, we can consider the complex topological relationships between samples, aiming to push the intra/inter-class nodes close/disperse for clustering improvement.

For the clustering task, the selection strategy for positive and negative samples is as follows: positive samples are defined as the cluster centroid of the sample and its own augmented view, while negative samples are the centroids of other clusters. For z_i, the clustering task loss is

$$\mathcal{L}_{CT(i)} = -\frac{1}{2}\sum_{j=1}^{2}\log\left(\frac{e^{\mathcal{S}\left(\left(z_i^1, z_i^2\right)/\tau\right)} + e^{\mathcal{S}\left(\left(z_i^j, c_i\right)/\tau\right)}}{e^{\mathcal{S}\left(\left(z_i^1, z_i^2\right)/\tau\right)} + \sum_k e^{\mathcal{S}\left(\left(z_i^j, c_k\right)/\tau\right)}}\right), \tag{11}$$

where c_i is the pseudo-centroid corresponding to z_i. Pulling nodes closer to their centroids and pushing them away from other centroids is beneficial to accelerate a clear clustering distribution. The overall loss of clustering contrastive task is

$$\mathcal{L}_{CT} = \frac{1}{n_{CT}}\sum_{i=1}^{n} v_i \ell_{CT(i)}. \tag{12}$$

By minimizing $\mathcal{L}_{CT}$, the node features with sufficient discrimination after discrimination task learning further show the clustering structure and facilitates clearer clustering separation.

Through the above task transition, the positive and negative sampling strategy gradually changes from considering only the node itself to mining the cluster structure, thus leveraging the clustering information to guide contrastive learning. The progression from easy to challenging tasks enables our method to consistently learn clustering-oriented discriminative features.

3.5 Joint Loss and Optimization

Combining Eqs. (3), (9), and (11) , the joint loss is

$$\mathcal{L} = \alpha\mathcal{L}_{DT} + \beta\mathcal{L}_{CT} + \gamma\mathcal{L}_{EN}, \tag{13}$$

where α, β and γ are hyper-parameters.

The loss of the model can be considered as a function of the model parameters $\mathbf{W}$ and the indicator vector v, which can be expressed as $\mathcal{L} = g(\mathbf{W}, v)$. To optimize the objective function, we use an alternate optimization algorithm to iteratively update both. Specifically, v is first initialized as an all-zero vector, which means that all nodes perform the discrimination task at the beginning of the model training. Perform the following two steps alternately until the final iteration.

Firstly, fix v^t and solve $\mathbf{W}^{t+1}$ by

$$\begin{aligned}\mathbf{W}^{t+1} = \arg\min_{\mathbf{W}^t} \big[&\alpha\mathcal{L}_{CT}(v^t, \mathbf{W}^t) + \beta\mathcal{L}_{DT}(v^t, \mathbf{W}^t) \\ &+ \gamma\mathcal{L}_{EN}(\mathbf{W}^t)\big].\end{aligned} \tag{14}$$

The model parameters $\mathbf{W}^{t+1}$ can be solved using the Adam optimizer.

Secondly, fix $\mathbf{W}^t$ and solve indicator vector v^{t+1} according to the cluster entropy E

$$v^{t+1} = \arg\min_{v^t}\sum_{i=1}^{N} v_i^t E_i^t, \text{ s.t. } \|v^t\|_1 = n_{CT}^t, \tag{15}$$

where n_{CT}^t initially starts at 0 and increases with the curriculum pace ε. $\|v\|_1$ is the L_1 norm of the vector. As n_{CT} increases, the indicator vector v will eventually become an all-ones vector, meaning that all nodes will participate in the clustering task.

4 Experiments

4.1 Benchmark Datasets

To substantiate the efficiency of the CurGL, fix publicly accessible real-world datasets are adopted as benchmarks, including CORA, UAT, PUBMED, AMAP, and AMAC. The datasets are collected from a range of domains such as air traffic, academic citation, and shopping networks. Further details regarding these datasets are shown in Table 1.

Dataset	Samples	Edges	Dimensions	Classes
CORA	2708	5429	1433	7
UAT	1190	13599	239	4
AMAP	7650	119081	745	8
AMAC	13752	245861	767	10
PUBMED	19717	44438	500	3

Table 1: Descriptions of real-world datasets.

4.2 Evaluation Metrics

The clustering result is evaluated with three well-known metrics, including Accuracy (ACC), Normalized Mutual Information (NMI), and Average Rand Index (ARI). All metrics are positively correlated with clustering performance, and the range is [0, 1].

4.3 Comparison with Competitors

Ten state-of-the-art node clustering algoritms are selected for comparative analysis. This evaluation encompasses a range of approaches, from the traditional k-Means algorithm to advanced GCN-based deep models, such as GAE [Kipf and Welling, 2016b], DAEGC [Wang *et al.*, 2019], and SDCN [Bo *et al.*, 2020], as well as contrastive learning-based techniques including GCA [Zhu *et al.*, 2021], SCAGC [Xia *et al.*, 2022], CCGC [Yang *et al.*, 2023], DCGL [Chen *et al.*, 2024], DCRN [Liu *et al.*, 2022b] and HSAN [Liu *et al.*, 2023].

Setups. The k-Means algorithm only utilizes the original node attributes as input. In contrast, other baseline methods use both the original node attributes and the topological graph. The hyper-parameters for each competitor are configured according to the recommendations provided in the original papers. For the proposed CurGL, we use the adaptive hyper-parameter selection, which means $\alpha = \frac{\|v\|_1}{N}$ and $\beta = 1-\alpha$. Additionally, a parameter grid search is conducted for γ. The advantage of this approach is that α will increase

Dataset	Metric	k-Means	GAE	DAEGC	SDCN	GCA	SCAGC	DCGL	DCRN	CCGC	HSAN	CurGL
CORA	ACC	26.27	63.80	70.43	50.70	53.62	73.45	64.77	61.93	73.88	77.07	**78.66**
	NMI	34.68	47.64	52.89	33.78	46.87	57.43	48.67	45.13	56.45	59.21	**60.24**
	ARI	19.35	38.00	49.63	25.76	30.32	52.24	36.20	33.15	52.51	57.52	**60.48**
UAT	ACC	42.47	56.34	52.29	52.25	51.15	53.24	46.81	49.92	56.34	56.04	**56.74**
	NMI	22.39	20.69	21.33	21.61	23.47	26.96	18.95	24.09	**28.15**	26.99	27.27
	ARI	15.71	18.33	20.50	21.63	20.52	22.49	16.49	17.17	25.52	25.22	**25.85**
AMAP	ACC	36.53	42.03	60.14	71.43	69.51	75.25	62.13	79.94	77.25	77.02	**80.16**
	NMI	19.31	31.87	58.03	64.13	60.70	67.18	57.26	**73.70**	67.44	67.21	72.21
	ARI	12.61	19.31	43.55	51.17	49.09	56.86	42.21	63.69	57.99	58.01	**63.75**
AMAC	ACC	36.44	43.14	49.26	54.12	54.92	58.43	OOM	OOM	53.57	OOM	**67.79**
	NMI	16.64	35.47	39.28	39.90	44.36	49.92	OOM	OOM	34.22	OOM	**55.26**
	ARI	28.08	27.06	35.29	28.84	35.61	38.29	OOM	OOM	32.42	OOM	**54.13**
PUBMED	ACC	43.83	62.09	68.73	59.21	69.51	72.42	OOM	OOM	42.58	OOM	**72.47**
	NMI	15.05	23.84	28.26	19.65	31.13	35.13	OOM	OOM	21.87	OOM	**37.02**
	ARI	11.43	20.62	29.84	17.07	30.85	34.19	OOM	OOM	21.23	OOM	**36.07**

Table 2: Node clustering performance (%) of nine methods on five datasets. The optimal and sub-optimal results are decorated with bold and underline, respectively. 'OOM' means out-of-memory.

Dataset	Metric	wo/CL	wo/CE	wo/CUd	wo/CUc	CurGL
CORA	ACC	77.06	77.21	72.60	71.52	**78.66**
	NMI	56.43	57.29	54.29	53.34	**60.24**
	ARI	56.31	58.05	51.13	43.35	**60.48**
UAT	ACC	55.46	55.13	48.74	50.92	**56.74**
	NMI	26.41	27.09	25.36	22.99	**27.27**
	ARI	24.41	25.24	15.17	19.65	**25.85**
AMAP	ACC	77.28	78.95	78.17	77.49	**80.16**
	NMI	67.28	71.33	70.05	69.60	**72.21**
	ARI	57.97	62.43	59.37	58.95	**63.75**
AMAC	ACC	65.58	67.37	59.70	58.38	**67.79**
	NMI	55.11	55.77	52.58	55.20	**55.26**
	ARI	48.06	51.21	51.42	41.00	**54.13**
PUBMED	ACC	63.83	71.91	61.58	67.25	**72.47**
	NMI	29.39	36.17	33.68	30.44	**37.02**
	ARI	26.06	35.23	29.01	27.94	**36.07**

Table 3: Node clustering performance (%) of ablation study. The optimal result are shown in bold.

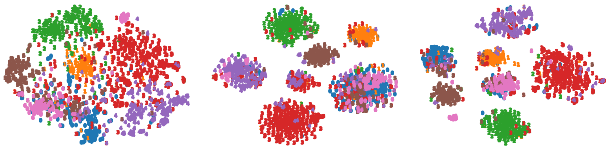
as the number of nodes involved in the clustering task grows, while β for the discrimination task will correspondingly decrease.

To ensure a fair comparison, each algorithm is executed 10 times to report the average. All deep models are trained with a NVIDIA RTX-4090 GPU.

Performance Comparison. Table 2 displays the average clustering performance of all algorithms. In general, the proposed CurGL outperforms other advanced methods, and achieves the best clustering results on all datasets, which indicates the practicability of CurGL on various graph clustering scenarios. From the experimental results, we also summarize the following viewpoints. Firstly, all GCN-based methods surpass k-Means on attributed graph clustering, which manifests the advantage of GCN on graph data mining. GCN-based deep models process the node attributes and the topological structure information simultaneously, so as to detect the internal data distribution more precisely. Secondly, benefiting from the efficient data augmentation techniques, GCN-based models outperform the traditional graph auto-encoder framework. The augmented view provide a more extensive semantic space for presentation learning, so as to improve clustering. Thirdly, compared to contrastive learning-based baselines, CurGL introduces the clustering guidance to improve the graph augmentation, which makes the augmented views more suitable for the downstream clustering task. Fourthly, unlike existing clustering-oriented methods such as CCGC, SCAGC, and HSAN, the proposed CurGL adjusts the contrastive learning task based on the learning state of the samples, alleviating the issue of unreliable pseudo-labels in the early stages.

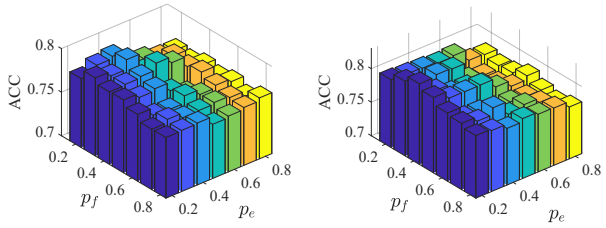
4.4 Ablation Study and Analysis

In this part, the ablation study is conducted to verify the effectiveness of the new mechanisms. Three variants of CurGL are designed, including wo/CL, wo/CE, wo/CUd and wo/CUc. Specifically, in wo/CL, the cluster-friendly graph augmentation is substituted with random data augmentation. In wo/CE, the clustering entropy loss is suspended. Furthermore, in wo/CUd and wo/CUc, the curriculum learning mechanism is suspended, allowing nodes to execute fixed contrastive learning during model training. wo/CUd represents that all nodes only perform the discrimination task, and wo/CUc represents that all nodes only perform the clustering task. Table 3 gives the ablation comparison on five datasets. It can be seen that CurGL still presents the best clustering scores, which proves the effects of the new modules on graph clustering. Furthermore, Fig. 2 shows the visualization results of the embeddings. CurGL obtains the best sample distribution with a clear cluster structure. The structure learned by wo/CUc is unclear



(a) wo/CUc (b) wo/CUd (c) CurGL

Figure 2: 2D Visualization of learned embeddings on Cora dataset. For better observation, only the first 100 samples of each class are selected.



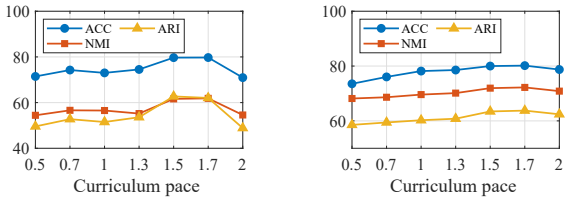
(a) COAR-ACC (b) AMAP-ACC

Figure 3: Clustering performance of CurGL with a different edge deletion probability p_e and feature masking probability p_f.

because only discriminative features of samples are learned. wo/CUd fails to capture the discriminative features, so some inter-class samples are grouped incorrectly.

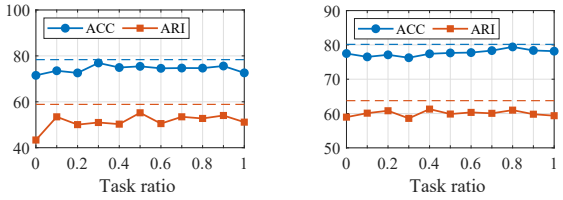
Impact of Augmentation probability. The augmentation probability determines the difference between the two views during contrastive learning. The edge deletion probability p_e and feature masking probability p_f control the intensity of structure and feature augmentation respectively. Fig. 3 shows the impact of different combinations of these two factors on clustering performance. It can be observed that when the augmentation probability is low, the data differences between the two views are very small. In this case, the selection space for positive and negative samples is limted, making it difficult to learn robust representations. On the other hand, when the augmentation probability is too high, the augmented views are excessively perturbed, where critical structures and features are damaged and original semantic information is lost.

Impact of Curriculum Pace. The curriculum pace ε determines the learning speed. When $\varepsilon = 1$, the model iterates to the last epoch, and exactly all nodes are transferred to the clustering task. When $\varepsilon > 1$, all nodes are transferred to the clustering task before the end of training. Fig. 4 shows the effect of curriculum pace. The clustering performance reaches the optimal when $1 < \varepsilon < 2$. The results are consistent with our original intention, as it tries to perform the clustering task for a period of epochs after all nodes complete the discrimination task. Too large curriculum pace also leads to poor clustering, since the sample is transferred to the clustering task prematurely without a discriminative embedding. The experimental result testifies the feasibility of multi-task contrastive scheme.



(a) CORA (b) AMAP

Figure 4: Effect of curriculum pace on clustering performance.



(a) CORA (b) AMAP

Figure 5: Clustering performance of CurGL with a fixed task ratio. The dotted line represents the performance with automatic task ratio.

Impact of Task Ratio. We remove the curriculum learning mechanism and fix the sample ratio of the two tasks to investigate the influence. When the task ratio is 0, all nodes only perform the discrimination task during the entire model training. When the task ratio is 1, all nodes engage in the complex clustering task. The results are visualized in Fig. 5. Obviously, relying solely on either the discrimination task or clustering task yields sub-optimal performances. The discriminanation task neglects the clustering guidance, and the clustering task suffer from the accumulation of adverse noise in pseudo-labels. Compared to performing a single task (i.e., task ratio is 0 or 1), multi-task learning with a fixed sample ratio can improve clustering effects, but it is still not as effective as CurGL with adaptive task allocation. To sum up, the adaptive adjustment of discrimination and clustering tasks is beneficial to learning clustering-friendly graph embedding.

5 Conclusion

In this paper, we establish a clustering-guided Curriculum Graph contrastive Learning (CurGL) framework. Clustering entropy is defined based on the embeddings to serve as the clustering guidance. After that, a clustering-friendly augmentation strategy is developed for structure-/feature-level graph augmentation, which avoids the noise brought by random augmentation. In addition, the proposed multi-task curriculum learning scheme performs contrastive learning on discrimination task in the early stage, and turns to clustering task in the late stage. The flexible transition strategy adjusts the sample selection strategy adaptively during the training process, and is more suitable for data with complex distribution. The efficiency of our method has been validated by a series of comprehensive experiments. In the future, we plan to extend CurGL to multi-view graph learning.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No:2022ZD0160803).

References

[Bengio *et al.*, 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[Bo *et al.*, 2020] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pages 1400–1410, 2020.

[Chen and Li, 2022] Mulin Chen and Xuelong Li. Entropy minimizing matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9209–9222, 2022.

[Chen *et al.*, 2024] Mulin Chen, Bocheng Wang, and Xuelong Li. Deep contrastive graph learning with clustering-oriented guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11364–11372, 2024.

[Cirik *et al.*, 2016] Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. Visualizing and understanding curriculum learning for long short-term memory networks. *arXiv preprint arXiv:1611.06204*, 2016.

[Cui *et al.*, 2020] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 976–985, 2020.

[Fang *et al.*, 2023] Taoran Fang, Zhiqing Xiao, Chunping Wang, Jiarong Xu, Xuan Yang, and Yang Yang. Dropmessage: Unifying random dropping for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4267–4275, 2023.

[Graves *et al.*, 2017] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. Pmlr, 2017.

[Hacohen and Weinshall, 2019] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR, 2019.

[Han *et al.*, 2018] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020.

[Hjelm *et al.*, 2019] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Ji *et al.*, 2021] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[Jiang *et al.*, 2018] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[Kipf and Welling, 2016a] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Kipf and Welling, 2016b] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[Li *et al.*, 2023] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data*, 17(1):1–21, 2023.

[Lin *et al.*, 2022] Shuai Lin, Chen Liu, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric Xing, and Xiaodan Liang. Prototypical graph contrastive learning. *IEEE transactions on neural networks and learning systems*, 35(2):2747–2758, 2022.

[Liu *et al.*, 2022a] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*, pages 1392–1403, 2022.

[Liu *et al.*, 2022b] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7603–7611, 2022.

[Liu *et al.*, 2023] Yue Liu, Xihong Yang, Sihang Zhou, Xinwang Liu, Zhen Wang, Ke Liang, Wenxuan Tu, Liang Li, Jingcan Duan, and Cancan Chen. Hard sample aware network for contrastive deep graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 8914–8922, 2023.

[Majeed and Rauf, 2020] Abdul Majeed and Ibtisam Rauf. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, 5(1):10, 2020.

[Matiisen *et al.*, 2019] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum

learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740, 2019.

[Mo *et al.*, 2022] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7797–7805, 2022.

[Newman *et al.*, 2002] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl_1):2566–2572, 2002.

[Platanios *et al.*, 2019] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019.

[Spitkovsky *et al.*, 2010] Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 751–759, 2010.

[Tay *et al.*, 2019] Yi Tay, Shuohang Wang, Luu Anh Tuan, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. *arXiv preprint arXiv:1905.10847*, 2019.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Veličković *et al.*, 2018] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax, 2018.

[Wang and Qi, 2022] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5549–5560, 2022.

[Wang *et al.*, 2019] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. *arXiv preprint arXiv:1906.06532*, 2019.

[Wang *et al.*, 2021a] Qianqian Wang, Jiafeng Cheng, Quanxue Gao, Guoshuai Zhao, and Licheng Jiao. Deep multi-view subspace clustering with unified and discriminative learning. *IEEE Transactions on Multimedia*, 23:3483–3493, 2021.

[Wang *et al.*, 2021b] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu. Generative partial multi-view clustering with adaptive fusion and cycle consistency. *IEEE Transactions on Image Processing*, 30:1771–1783, 2021.

[Wang *et al.*, 2021c] Qianqian Wang, Huanhuan Lian, Gan Sun, Quanxue Gao, and Licheng Jiao. icmsc: Incomplete cross-modal subspace clustering. *IEEE Transactions on Image Processing*, 30:305–317, 2021.

[Wang *et al.*, 2021d] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[Xia *et al.*, 2022] Wei Xia, Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao. Self-consistent contrastive attributed graph clustering with pseudo-label prompt. *IEEE Transactions on Multimedia*, 25:6665–6677, 2022.

[Xu *et al.*, 2024] Yanchen Xu, Siqi Huang, Hongyuan Zhang, and Xuelong Li. Why does dropping edges usually outperform adding edges in graph contrastive learning? *arXiv preprint arXiv:2412.08128*, 2024.

[Yang *et al.*, 2023] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 10834–10842, 2023.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[Zbontar *et al.*, 2021] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[Zhang *et al.*, 2024] Hongyuan Zhang, Yanchen Xu, Sida Huang, and Xuelong Li. Data augmentation of contrastive learning is estimating positive-incentive noise. *arXiv preprint arXiv:2408.09929*, 2024.

[Zhao *et al.*, 2019] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21(9):3848–3858, 2019.

[Zhao *et al.*, 2021] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. Graph debiased contrastive learning with joint representation clustering. In *IJCAI*, pages 3434–3440, 2021.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, pages 2069–2080, 2021.