MocGCL: Molecular Graph Contrastive Learning via Negative Selection

Jinhao Cui*, Heyan Chai*, Yanbin Gong†, Ye Ding‡, Zhongyun Hua*, Cuiyun Gao*, Qing Liao*§✉
*Harbin Institute of Technology Shenzhen, Shenzhen, China
†The Hong Kong University of Science and Technology, Hong Kong, China
‡Dongguan University of Technology, Dongguan, China
§Peng Cheng Laboratory, Shenzhen, China
{cuijinhao, chaiheyan}@stu.hit.edu.cn, ygongae@connect.ust.hk, dingye@dgut.edu.cn
{huazhongyun, gaocuiyun, liaoqing}@hit.edu.cn

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191518

***Abstract*—Molecular classification benefits a lot from the recent success of graph contrastive learning (GCL) which pulls positive samples close and pushes the negative samples apart. GCL methods generate negative and positive samples via graph augmentation. Due to the structural corruption caused by graph augmentation, not all generated negative samples retain discriminative semantics. However, existing GCL methods ignore the difference between negative samples and hold an assumption that the importance of all negative samples is the same, leading to degraded performance of molecular classification. To address this issue, in this paper, we propose a novel molecular graph contrastive learning model (MocGCL) by selecting more useful negative samples to improve the performance of molecular classification. Specifically, we first employ different encoders to generate positive samples to improve the diversity of positive samples. Then, we design negative generation to generate negative samples and define semantic integrity to measure the usefulness of generated negative samples. Moreover, we propose the novel negative selection to dynamically select the negative samples of more usefulness to improve the molecular representation. In addition, we improve the contrastive loss to adaptively adjust the distance between selected negative samples, which can preserve the distinctive properties of selected negative samples in sample space. Extensive experiments on six typical bioinformatics datasets demonstrate the effectiveness of our MocGCL compared to most state-of-the-art methods.**

***Index Terms*—Graph contrastive learning, molecular classification, self-supervised learning**

I. Introduction

Molecular classification is to determine whether some molecular graphs have certain properties [1]–[3], which has wide real-world applications, such as genetic classification [4], drug development [5], and cancer detection [6], [7]. The previous molecular classification methods focus on using extensive domain knowledge to identify some representative local molecular structures. These methods suffer huge computational costs and are insufficient for global structure analysis [8]–[10]. Recently, owing to the impressive representational power in various domains, graph neural networks (GNNs) [11]–[13] gain increasing attention in the field of molecular classification. The above methods are all trained in a supervised manner, which relies on sufficient fine-annotated molecules. However, labeling molecules incurs massive time costs and requests a lot of domain knowledge [14], [15].

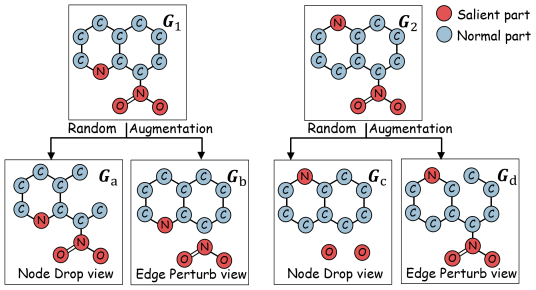

Fig. 1. The illustration of different qualities of generated samples. Molecular graphs G_a and G_b are positive pairs, and molecular graphs G_c and G_d are also positive pairs. Molecular graphs G_a, G_b, G_c, and G_d are treated as negative samples for other molecular graphs.

More recently, self-supervised graph contrastive learning (GCL) [16] methods achieve overwhelming accomplishments in the field of molecular classification. These methods mostly apply graph augmentation to generate positive and negative samples. To obtain the refined molecular representation to boost molecular classification without relying on fine annotations, GCL pulls the positive samples[1] close to each other and pushes negative samples[2] apart. Existing GCL methods [17]–[19] generate samples in a random fashion and select the prefabricated graph augmentation operations per dataset to improve the performance of GCL on molecular classification. However, molecular graphs contain a wealth of fine structural information. Thus, augmenting molecular graphs causes the generated samples to lose their salient semantics and be far away from their original molecular graphs.

As shown in Figure 1, G_a is generated from G_1 via node dropping graph augmentation, and G_b is generated from G_1 via edge perturbation. In the previous methods, G_a and G_b are employed without discrimination when these two samples are treated as negative samples. However, G_a inherits the salient properties from the original graph, but the salient properties in G_b are corrupted. The generated samples G_a with more salient

[1]Samples generated from the same molecule (anchor molecule) are positive samples for each other

[2]Samples generated from other molecules are treated as negative samples for anchors

parts can provide abundant molecular attributes to facilitate molecular representation learning. Thus, G_a is more useful than G_b in GCL. Similarly, G_d is more useful than G_c. Employing G_b and G_c as negative samples leads to inefficiency in contrastive learning. Therefore, generated samples have different qualities. Selecting more useful generated samples can improve the performance of GCL.

The present methods mostly ignore that some negative samples which preserve less original molecular structure cannot contribute the distinctive semantics of original molecules to positive samples. We argue that treating negative samples equally leads to performance degradation, which cannot provide sufficient molecular attributes for molecular representation learning in GCL. Therefore, we should select negative samples that can be more useful to positive samples, so as to help the model capture refined molecular representation.

In this paper, we propose a novel molecular graph contrastive learning (MocGCL) to improve the performance of molecular classification by selecting negative samples. Specifically, to improve the sample diversity of positive samples and reduce the significant semantics corruption, we first use different encoders to generate positive sample representation. Then, we design a negative generation to obtain negative samples and measure the usefulness of negative samples via semantic integrity. Moreover, a negative selection module is proposed to dynamically select more useful negative samples in each training iteration based on the ranking of semantic integrity. In addition, we improve the contrastive loss to preserve the distinctive properties of negative samples in sample space. We highlight the major contribution of this paper as follows:

- We propose a novel molecular graph contrastive learning (MocGCL) that can select more useful negative samples to facilitate the learning of molecular representation to improve the performance of molecular classification.
- We propose the negative selection by considering semantic integrity to select negative samples with more usefulness, so as to obtain richer molecular representation.
- Experiments on various datasets demonstrate that MocGCL with selecting negative samples can improve the performance of molecular classification compared to several state-of-the-art methods.

II. Related Work

A. Supervised Molecular classification

Early molecular classification methods [2], [10], [20], [21] typically employ atoms as vertices, and bonds as edges to learn the graph-based representation of molecules. For example, WL [20] uses the number of different atoms labels of subtree kernels as the feature vectors of molecular graphs, and DGK [21] defines the representation of molecules by structural similarity. Recently, GNNs have shown their powerful ability to learn the graph-based representations of molecules. For example, GCN [12] implements molecular graph convolution by using the Laplace transform, which reduces the heavy computational costs. GAT [13] focuses more on the message-passing process between atoms and neighbors rather than on the molecular structure, which leverages attention mechanisms to aggregate the neighbors' information with different weights. GIN [11] utilizes GNNs to construct a network structure with the same strength of expressiveness as the Weisfeiler-Lehman (WL) test [11]. The aforementioned methods are all trained in a supervised manner, which relies on the sufficient fine-annotated molecules. However, labeled molecular graphs are scarce, and labeling molecular graphs incurs additional time overhead. Our MocGCL uses a self-supervised training manner that can be independent of the fine-annotated molecules.

B. Graph Contrastive Learning

More recently, tremendous researches focus on graph self-supervised learning [16], [22], [23] that can obtain information from unlabeled molecular graphs. Among them, graph contrast learning (GCL) can maximize the agreement between positive samples compared with negative samples to capture molecular representations. Typically, GraphCL [17] demonstrates learning molecular representation by pretraining can help molecular classification. JOAO [18] proposes a unified bi-level optimization framework that allows data augmentation methods to be selected during different training phases based on the molecular datasets. MoCL [19] incorporates bioinformatics domain knowledge into data augmentation to avoid altering semantics. SimGRACE [24] employs encoder perturbations instead of data augmentation to generate positive samples and negative samples. However, previous GCL methods ignore the fact that generated negative samples have different contributions to positive samples. Selecting discriminative negative samples that are more useful for positive samples can improve the performance of molecular classification.

III. Methodology

In this section, we present MocGCL in detail. As sketched in Figure 2, we first introduce the positive and negative generation, then followed by the negative selection. Finally, we introduce the improved contrastive loss of MocGCL.

Graph G is represented as $G = (V, E, A)$, where $V = \{v_1, v_2, \cdots, v_n\}$ denotes the node set, E denotes the set of edges, $A \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix, and n is the number of nodes. The negative sample generated from G is represented as $\hat{G} = (\hat{V}, \hat{E}, \hat{A})$.

A. Positive Generation

The previous GCL methods mostly apply graph augmentation operations to generate positive samples. However, augmenting molecular graphs corrupts the salient properties, and positive samples may have overlapping structures. To reduce the semantic corruption and improve the sample diversity of positive samples, we utilize the GNNs [11] encoder and its momentum-update version [25] to generate positive samples at the representation level. The GNNs encoder f_q can be formulated as,

$$\boldsymbol{h}_G = f_q(G; \theta_q), \tag{1}$$

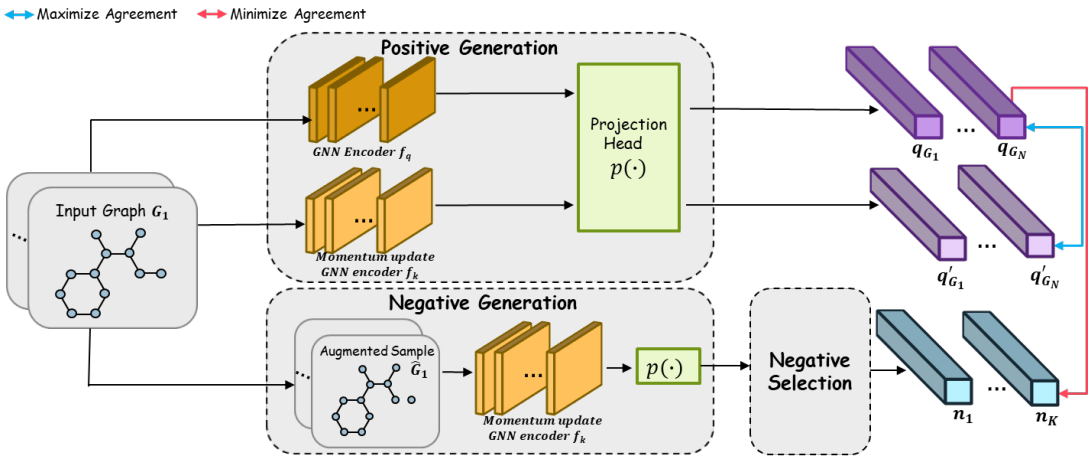

Fig. 2. The overall illustration of the MocGCL architecture.

where $\boldsymbol{h}_G$ is the representation of molecular graph G, and θ_q is the parameter of GNNs encoder f_q. The momentum-update version GNNs encoder f_k can be formulated as,

$$\boldsymbol{h}_G^{'} = f_k(G;\theta_k), \tag{2}$$

where $\boldsymbol{h}_G^{'}$ is also the representation of molecular graph G, and θ_k is the parameter of momentum-update version GNNs encoder f_k. The process of momentum-update strategy can be described as,

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q, \tag{3}$$

where $m \in [0,1)$ is a momentum coefficient. During the training, θ_q is updated by back-propagation, but θ_k is only updated from θ_q by the momentum-update Eq.(3).

Moreover, we adopt a two-layer MLP $p(\cdot)$ as the projection head to map the representations to a task-based latent space, which can enhance the performance of molecular classification [26]. The projection head can be described as,

$$\boldsymbol{q}_G = p(\boldsymbol{h}_G), \boldsymbol{q}_G^{'} = p(\boldsymbol{h}_G^{'}), \tag{4}$$

where $\boldsymbol{q}_G$ and $\boldsymbol{q}_G^{'}$ are the final positive samples representation after projecting, and these two representations are obtained from the same graph G.

B. Negative Generation

In GCL, samples generated from other molecules are treated as negative samples for anchors. In this subsection, we introduce the operations of generating negative samples.

Graph Augmentation: Following the GraphCL [17], we apply two graph augmentation operations in a random fashion to generate negative samples: (1). Node Dropping: randomly drop a node along with its connections. (2). Edge Perturbation: remove an edge randomly. We randomly choose the above two operations for the molecular graph G to generate one negative sample $\hat{G}$. To keep the negative samples' representation consistent in the latent space, we obtain the representations of negative samples by the momentum-update encoder via Eq.(2) and Eq.(4), which can be described as,

$$\boldsymbol{h}_{\hat{G}} = f_k(\hat{G};\theta_k), \boldsymbol{q}_{\hat{G}} = p(\boldsymbol{h}_{\hat{G}}), \tag{5}$$

where $\boldsymbol{h}_{\hat{G}}$ is the representation of negative sample $\hat{G}$, and $\boldsymbol{q}_{\hat{G}}$ is the final representation of the negative sample $\hat{G}$ after projecting. Subsequently, we use $\mathbf{n}_i$ to denote $\boldsymbol{q}_{\hat{G}_i}$ to easily distinguish between positive and negative samples.

C. Negative Selection

Applying the above two graph augmentation operations alters the molecular semantics. Some generated negative samples with less original salient structure cannot contribute sufficient molecular semantics to positive samples to obtain refined molecular representation. To tackle this problem, we design a novel negative selection to select more useful generated negative samples.

Semantic Integrity Calculation: To identify the negative samples of more usefulness, we define the semantic integrity $\psi_{\hat{G}}$ of the sample $\hat{G}$ by calculating the semantic importance of each node and edge in the original molecular graph G. To measure the importance of node v_i to molecular semantics, we first define the initial semantic importance $\varphi_{v_i}^0$, which can be described as,

$$\varphi_{v_i}^0 = d_{v_i}, \tag{6}$$

where d_{v_i} denotes the number of edges connecting node v_i. To further take the importance of both a node and its neighboring nodes into consideration, we apply eigenvector centrality iteration [27] to calculate the final semantic importance of nodes. The eigenvector centrality iteration can be described as:

$$\Phi_G(t) = A\Phi_G(t-1), \tag{7}$$

where $\Phi_G(t) = [\varphi_{v_1}^t, \varphi_{v_2}^t, \cdots, \varphi_{v_n}^t]^T$ is the semantic importance matrix of nodes after iterating t turns, and T means transpose of the matrix. Specifically, $\Phi_G(0) = [d_{v_1}, d_{v_2}, \cdots, d_{v_n}]^T$ denotes the degree matrix of graph G. $t \in \mathbb{N}$ denotes the number of iterations, and A is the adjacency matrix of graph G. The iteration Eq.(7) terminates in case that $\Phi_G(t)$ and $\Phi_G(t-1)$ are equal after normalization, and $\Phi_G(t)$ is the final semantic importance matrix of nodes in graph G.

Then, in the molecular graph G, we define the semantic importance of an edge as the weighted average of two adjacent nodes' semantic importance, which can be described as,

$$\varphi_{e_{ij}}^t = \frac{\varphi_{v_i}^t}{d_{v_i}} + \frac{\varphi_{v_j}^t}{d_{v_j}}, \tag{8}$$

where e_{ij} is the edge connecting v_i and v_j, degree d_{v_i} and d_{v_j} denote the number of edges connecting node v_i and node v_j respectively.

After obtaining the semantic importance matrix $\Phi_G(t)$, we utilize the semantic importance of both the nodes and edges to calculate the semantic integrity $\psi_{\hat{G}}$ of the generated negative sample $\hat{G}$. We present the calculation of semantic integrity in two cases according to the two graph augmentation operations: (1). Node Dropping: when we generate negative sample $\hat{G}$ by dropping a node v_k from molecular graph G, the semantic integrity $\psi_{\hat{G}}$ can be calculated as,

$$\psi_{\hat{G}} = \frac{\sum_{v_i \in V, i \neq k} \varphi_{v_i}^t}{\sum_{v_i \in V} \varphi_{v_i}^t}, \tag{9}$$

where φ_{v_i} is the semantic importance of node v_i in graph G, and V is the node set of G.

(2). Edge Perturbation: when we generate the negative sample $\hat{G}$ by deleting an edge e_{uv}, the semantic integrity $\psi_{\hat{G}}$ can be calculated as,

$$\psi_{\hat{G}} = \frac{\sum_{e_{ij} \in E, e_{ij} \neq e_{uv}} \varphi_{e_{ij}}^t}{\sum_{e_{ij} \in E} \varphi_{e_{ij}}^t}, \tag{10}$$

where $\varphi_{e_{ij}}$ is the semantic importance of edge e_{ij} in graph G, and E denotes the edges set of G.

To measure the usefulness of different generated negative samples, we need to compare the semantic integrity between generated samples in different two augmenting cases.

Theorem III.1. *For a given molecular graph G, the sum of the semantic importance of all nodes is constantly equal to the sum of the semantic importance of all edges, which can be described as,*

$$\sum_{e_{ij} \in E} \varphi_{e_{ij}}^t \equiv \sum_{v_i \in V} \varphi_{v_i}^t \tag{11}$$

Proof. According to Eq.(8), we prove the Theorem III.1:

$$\begin{aligned}
\sum_{e_{ij} \in E} \varphi_{e_{ij}}^t &= \sum_{e_{ij} \in E} \frac{\varphi_{v_i}^t}{d_{v_i}} + \frac{\varphi_{v_j}^t}{d_{v_j}} \\
&= \underbrace{\frac{\varphi_{v_1}^t}{d_{v_1}} + \cdots + \frac{\varphi_{v_1}^t}{d_{v_1}}}_{d_{v_1}} + \cdots + \underbrace{\frac{\varphi_{v_n}^t}{d_{v_n}} + \cdots + \frac{\varphi_{v_n}^t}{d_{v_n}}}_{d_{v_n}} \\
&= \sum_{v_i \in V} d_{v_i} \frac{\varphi_{v_i}^t}{d_{v_i}} \\
&= \sum_{v_i \in V} \varphi_{v_i}^t,
\end{aligned}$$

□

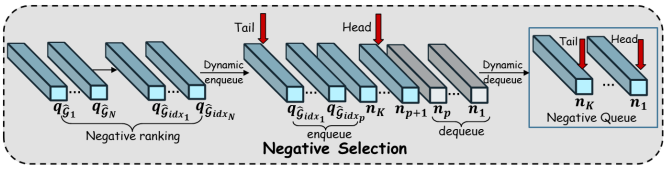

Fig. 3. The architecture of negative selection.

Theorem III.1 demonstrates that the maximum semantic integrity calculated by Eq.(9) is exactly equal to the maximum semantic integrity calculated by Eq.(10). In other words, by our definition of nodes and edges' semantic importance, the semantic integrity calculated in two different graph augmentation operations is equivalent. The random selection of two graph augmentation operations to generate negative samples does not introduce ambiguity in semantic integrity.

After calculating the semantic integrity, we select some negative samples with more usefulness by comparing semantic integrity. Figure 3 illustrates the scheme of our proposed negative selection. We dynamically select top-p useful negative samples and maintain a negative queue $\mathcal{Q}$ of size K as the negative set. The negative queue $\mathcal{Q}$ can be described as,

$$\mathcal{Q} = \{\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_i, \cdots, \mathbf{n}_K\} \tag{12}$$

Where $\mathbf{n}_i$ is the selected negative sample $\hat{G}_i$'s representation $\mathbf{q}_{\hat{G}_i}$ obtained via the Eq.(5). For ease of description, we rename the representation of negative sample $\hat{G}_i$ as $\mathbf{n}_i$. The negative queue can reuse the negative samples generated from the previous mini-batches.

Dynamic Entry: In the previous methods CSSL [23] and MoCo [25], the number of queue entries per mini-batch p is constant, e.g., mini-batch size. However, with the updating of networks, the representation of negative samples becomes close to each other in some mini-batches. We argue that negative samples of these mini-batches lose their rich molecular semantics at the representation level, so this decreases the number of negative samples with more usefulness. Hence, we dynamically adjust p queue entries during the training. Specifically, to measure the distance of negative samples in l-th mini-batch, we define the clustering degree by the average of the cosine similarity between negative samples,

$$C^l = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} cos(\mathbf{n}_i, \mathbf{n}_j), \tag{13}$$

where C^l is the clustering degree, $cos(\cdot)$ is the cosine similarity, N is the number of anchor molecular graphs, and $\mathbf{n}_i$ is the representation of the negative sample. Then, we calculate the number of queue entries p through the comparison of the clustering degree between the current mini-batch and the neighboring mini-batch. The number of queue entries p can be described as,

$$p = \begin{cases} \frac{\sum_{i=l-T}^{l-1} C^i}{\sum_{i=l-T}^{l-1} C^i + (T-1)C^l} \cdot N & , l > T \\ N & , l \leq T, \end{cases} \tag{14}$$

where p is the number of queue entries, and T is a predefined parameter. Especially, when $l \leq T$, all negative samples in the current mini-batch enter the negative queue to avoid the cold boot issue.

Negative ranking: To select the top-p useful negative samples, we rank the generated negative samples according to semantic integrity. The negative samples with higher semantic integrity can be prioritized into the negative queue in each training round. In the current l-th mini-batch, $\mathcal{G} = \{G_1, G_2, \cdots, G_N\}$ is the anchor molecular graph set, and $\hat{G}_i$ is one negative sample generated from the corresponding anchor G_i. N is the number of anchor molecular graphs. The selection of negative samples can be described as,

$$idx = rank(\{\psi_{\hat{G}_1}, \psi_{\hat{G}_2}, \cdots, \psi_{\hat{G}_N}\}, p), \tag{15}$$

where $rank(\cdot)$ is the operation of negative samples ranking, idx returned by $rank(\cdot)$ denotes the subscript of selected negative samples, p is the number of queue entries calculated via Eq.(14), and $\psi_{\hat{G}_i}$ is the semantic integrity of negative sample $\hat{G}_i$ calculated by the Eq.(9) or Eq.(10). The set of selected negative samples can be presented as $\{\mathbf{n}_{idx[1]}, \mathbf{n}_{idx[2]}, \cdots, \mathbf{n}_{idx[p]}\}$. After the current training round finishes, these selected negative samples enter the negative queue, and the same number of negative samples of the oldest mini-batch in the negative queue is removed.

D. Contrastive Loss

We take the pretrain and finetune scheme to train our model. We first pretrain our MocGCL to enforce the agreement between positive samples $\boldsymbol{q}_G$ and $\boldsymbol{q}_G^{'}$ compared with the negative sample set $\mathcal{Q} = \{\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_K\}$ to pretrain the encoder f_q and the projection $p(\cdot)$. Following the InfoNCE [28] used in the previous methods [17]–[19], [24], we adjust the contrastive loss with the attention mechanism to keep the distinctive properties of selected negative samples in sample space. The improved contrastive loss $\mathcal{L}_{\mathcal{P}}$ can adaptively adjust the distance between negative samples' representations to improve the uniformity of negative samples, which can be described as,

$$\mathcal{L}_{\mathcal{P}} = \frac{-1}{N}\sum_{i=1}^{N} log\frac{exp(\mathbf{q}_{G_i}^T \mathbf{q}_{G_i}^{'})/\tau}{exp(\mathbf{q}_{G_i}^T \mathbf{q}_{G_i}^{'})/\tau + \sum_{k=1}^{K} a_{ik} \cdot exp(\mathbf{q}_{G_i}^T \mathbf{n}_k)/\tau}, \tag{16}$$

where N is the size of the mini-batch, K is the size of the negative queue Q, $\tau \in (0, 1]$ is the temperature parameter that can change the uniformity of negative samples' representations, a_{ik} is the attention coefficient which can be described as,

$$a_{ik} = \mathbf{q}_{G_i}^T \mathbf{n}_k. \tag{17}$$

After pretraining, we finetune the encoder f_q and the projection $p(\cdot)$ based on the molecular classification task and predict the molecular properties.

TABLE I
STATISTICS OF DATASETS.

Dataset	#Graphs	#Classes	#Avg.Edges	#Avg.Nodes
D&D [29]	1178	2	715.66	284.32
PROTEINS [30]	1113	2	72.82	39.06
ENZYMES [30]	600	6	62.14	32.63
NCI1 [31]	4110	2	32.30	29.87
MUTAG [32]	188	2	19.79	17.93
Mutagenicity [33]	4337	2	30.77	30.32

IV. EXPERIMENTS

In this section, we present experiments conducted to demonstrate the effectiveness of our MocGCL for molecular classification.

A. Experimental setups

Datasets: We use six typical bioinformatics datasets namely D&D [29], PROTEINS [30], ENZYMES [30], NCI1 [31], MUTAG [32], and Mutagenicity [33]. The datasets' statistics are summarized in Table I. We randomly split each dataset into three parts: 80% for the train set, 10% for the validation set, and 10% for the test set. Each full dataset without labels is used for pretraining, and the corresponding dataset with labels is used for finetuning. The random split is repeated 10 times, and the average performance with standard deviation is reported.

Baselines: We compare the following baselines to demonstrate the effectiveness of our MocGCL on molecular classification. Baselines can be divided into supervised methods and self-supervised methods. *The supervised methods* contain three categories: (1). Molecular graph kernel-based methods which focus on performing classification based on the similarity between molecules include Weisfeiler-Lehman Subtree Kernel(WL) [20] and Deep Graph Kernels(DGK) [13]. (2). Molecular graph neural network methods which both utilize the neural networks and molecular structure information include Graph Convolutional Network(GCN) [12] and Graph Isomorphism Network(GIN) [11]. (3). Molecular graph pooling methods which combine graph neural networks with pooling mechanisms include gPool [34] and EigenPooling [35]. *The self-supervised methods* which obtain molecular representation from unlabeled molecular graphs include InfoGraph [36], GraphCL [17], SUGAR [37], CSSL [23], SimGRACE [24], and FGCL [38].

Parameter settings: The common parameters for training the model are set as momentum coefficient $m = 0.9$, temperature parameter $\tau = 0.07$, dropout ratio = 0.5, and L2 norm regularization weight decay = 0.01. For NCI1 [31] and Mutagenicity [33], negative queue size $K = 2048$. For D&D [29], PROTEINS [30], ENZYMES [30] and MUTAG [32], negative queue size $K = 512$. For each dataset, the parameter T is predefined as the negative queue size divided by the size of the mini-batch to avoid the cold boot issue. We adopt GIN [11] as the GNNs encoders with 3 layers and 32 hidden dimensions.

TABLE II
SUMMARY OF EXPERIMENTAL RESULTS: "AVERAGE ACCURACY(%)±STANDARD DEVIATION (RANK)".

Method	Categories	Dataset						Average Rank
		ENZYMES	PROTEINS	D&D	NCI1	Mutagenicity	MUTAG	
WL [20]	Supervised	52.22±1.26	- [1]	76.44±2.35	76.65±1.99	80.32±1.71	82.05±0.36	8.4
DGK [13]		53.43±0.91	75.68±0.54	-	80.31±0.46	-	87.44±2.72	6.5
GCN [12]		49.63±3.27	75.17±3.63	73.26±4.46	76.29±1.79	76.40±0.61	80.20±2.86	11
GIN [11]		51.33±2.08	76.20±2.80	79.53±1.34	76.76±1.19	76.24±0.74	89.40±5.60	8
EigenPool [35]		64.67±0.24	78.84±1.06	78.63±1.36	77.24±0.96	80.11±0.73	78.9±0.49	6.5
gPool [34]		43.00±4.20	77.68±1.75	77.02±1.32	76.25±1.39	80.30±1.54	89.30±6.85	8.7
InfoGraph [36]	Self-Supervised	53.41±2.34	75.18±0.51	74.24±0.86	70.93±1.78	72.32±1.70	89.01±1.13	9.7
GraphCL [17]		50.35±2.16	78.57±1.54	81.23±0.56	80.44±0.89	76.88±1.43	88.91±1.63	6.8
SUGAR [37]		52.88±2.29	81.34±0.93	84.03±1.33	**84.39±1.63**	78.99±1.00	96.74±4.55	3.7
CSSL [23]		51.47±1.39	82.50±1.01	82.18±1.34	80.09±1.07	82.64±0.83	91.01±3.66	4.3
SimGRACE [24]		52.33±1.37	78.81±1.38	79.04±1.73	76.77±0.65	74.58±0.67	93.10±3.10	6.8
FGCL [38]		57.39±1.67	73.91±1.88	**85.29±0.76**	79.46±0.91	81.57±1.01	-	5
MocGCL(Ours)		**65.30±2.48**	**84.91±1.81**	84.77±1.52	83.19±1.29	**83.44±0.58**	**97.50±3.35**	**1.3**

TABLE III
ABLATION STUDY RESULTS: "ACCURACY(%)"

Model	Dataset				Average Acc
	NCI1	Mutag	ENZYMES	PROTEINS	
w/o SS	79.13	81.79	62.13	81.63	76.17
Full-Batch	78.61	81.11	63.27	81.85	76.21
Semi-Batch	79.28	82.11	63.33	82.07	76.69
Quarter-Batch	78.95	81.98	61.83	81.69	76.11
w/o $\mathcal{L}_{\mathcal{P}}$	80.26	83.18	63.46	83.04	77.49
MocGCL	**83.19**	**83.44**	**65.30**	**84.91**	**79.21**

B. Overall performance results

We evaluate our MocGCL on the six above-mentioned datasets. The experimental results are summarized in Table II where the best results are shown in bold. As illustrated in Table II, MocGCL outperforms baselines on four datasets and has the highest average accuracy on all datasets. Specifically, MocGCL outperforms the second-best self-supervised method FGCL [38] by 13.78% on the ENZYMES [30] dataset. This may be because MocGCL can capture refined molecular representation by selecting more useful negative samples so that the model is more conducive to multi-categories classification tasks. Besides, The possible reason for MocGCL falling behind FGCL and SUGAR [37] is that these two baselines both propose a pooling mechanism. The pooling mechanism can disentangle the graphs into hierarchical graphs in GCL, which makes these models perform better on the denser NCI1 and D&D datasets. Under similar dataset sizes, the NCI1 [31] and D&D [33] are denser than Mutagenicity [33] and PROTEINS [30] respectively. Hence, MocGCL outperforms FGCL and SUGAR methods on both the Mutagenicity and PROTEINS datasets. In general, Our MocGCL shows a significant improvement over the recently developed methods.

[1]The reported results of the baseline methods come from the initial publications ("-" means results and public code are both not available).

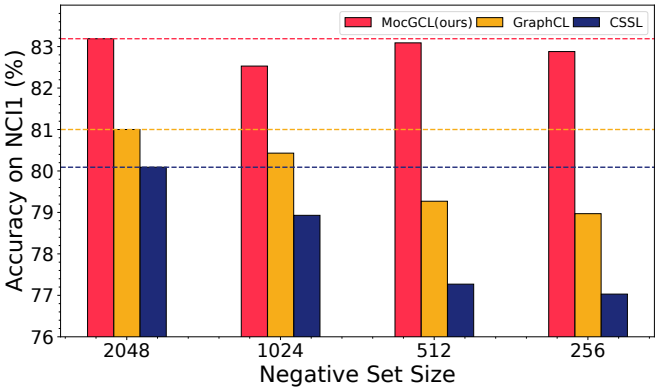

Fig. 4. The impact of different negative set sizes on NCI1 dataset.

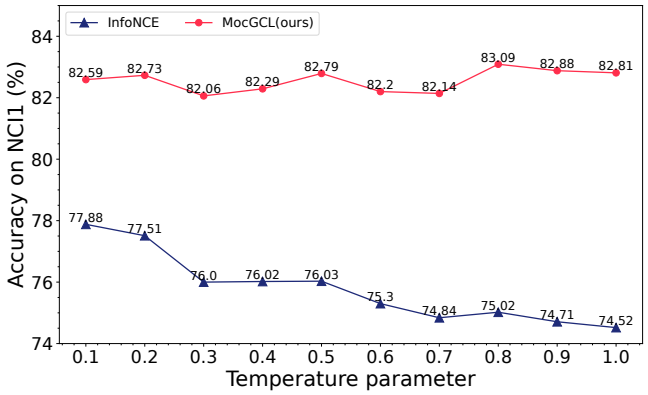

Fig. 5. Temperature parameter sensitivity analysis on NCI1 dataset.

C. Ablation study

In this subsection, table III presents the ablation study results on four datasets. To verify the validity of negative selection, we first perform MocGCL without selecting negative samples (w/o SS). Then, we investigate the effectiveness of

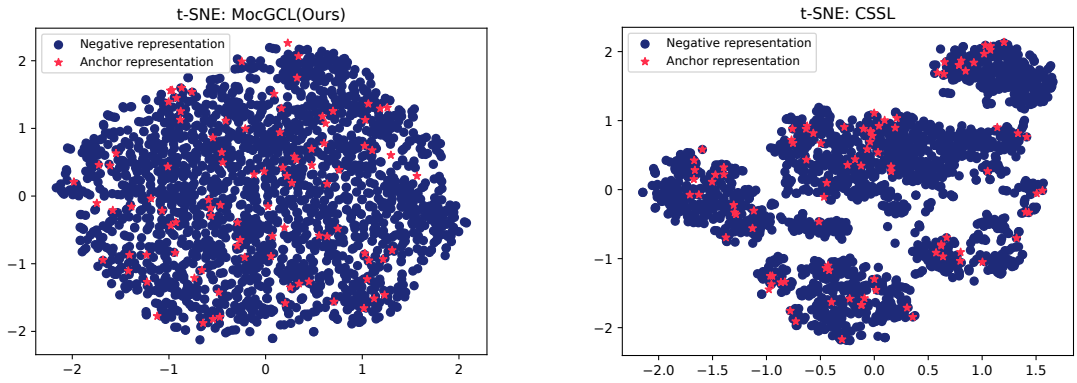

Fig. 6. t-SNE visualization on NCI1 dataset. Negative representations are obtained from the negative queue set and anchor molecular representations are obtained from the nearby six mini-batches before the end of pretraining.

dynamic entry by fixing p queue entries in each training round: (1) Full-Batch model: we fix the number of queue entries as the size of the full mini-batch. (2) Semi-Batch model: we fix the number of queue entries as the half size of the mini-batch. (3) Quarter-Batch model: we fix the number of queue entries as the quarter size of the mini-batch. In addition, we also perform MocGCL using InfoNCE loss instead of contrastive loss $\mathcal{L}_{\mathcal{P}}$ (w/o $\mathcal{L}_{\mathcal{P}}$). As shown in Table III, the proposed MocGCL achieves the best average accuracy of 79.21% on four datasets. The model without negative selection (w/o SS) degrades by at most 4.9% on all datasets substantially. This demonstrates that selecting more useful negative samples facilitates molecular representation learning in GCL. Fixing the number of queue entries also degrades the performance of MocGCL. This indicates that the dynamic entry mechanism can dynamically select more useful negative samples to improve the performance of GCL. Moreover, table III also shows that contrastive loss $\mathcal{L}_{\mathcal{P}}$ can improve the performance by at most 2.2% on molecular classification.

D. Negative set size analysis

Figure 4 presents the results of negative set size analysis on CSSL [23], GraphCL [17], and proposed MocGCL. As shown in Figure 4, when the negative set size equals 2048, all methods achieve the best performance, and our proposed MocGCL outperforms GraphCL by 2.4%. Moreover, the performance of both GraphCL and CSSL decreases as the negative set size decreases, but the performance of MocGCL remains around 83%. A smaller negative set size provides fewer negative samples, which further demonstrates that dynamic entry can improve the quality of negative samples to improve the performance of GCL on molecular classification.

E. Temperature parameter sensitive analysis

Figure 5 presents the impact of the temperature parameter on our proposed MocGCL. We respectively adjust the value of the temperature parameter in the contrastive loss $\mathcal{L}_{\mathcal{P}}$ and the InfoNCE [28]. As shown in Figure 5, the performance in MocGCL always outperforms that in InfoNCE for any of the temperature parameters. Moreover, the fluctuations in the InfoNCE's performance decrease significantly when changing the temperature parameter, but our proposed MocGCL is insensitive to temperature parameters. This indicates that our proposed MocGCL can keep the distinctive properties of selected negative samples in sample space without relying too much on the regulation of the temperature parameter.

F. Visualization

Figure 6 visualizes the distribution of negative samples' representation in CSSL [23] and our MocGCL by t-SNE on the NCI1 dataset. As shown in Figure 6, we note that the distribution of negative sample representation show clustering in CSSL but is uniformly distributed in MocGCL. Negative samples in CSSL are close to each other in the same cluster, which demonstrates some negative samples lose their distinctive properties. Too more negative samples staying in the same cluster causes negative sample redundancy. Contrary to that, MocGCL can keep the uniformity of negative samples. This indicates that MocGCL can select more kinds of negative samples which can contribute more molecular semantics to positive samples.

V. Conclusion

In this paper, we propose MocGCL, a novel molecular graph contrastive learning via negative selection to improve the performance of GCL on molecular classification. First, the positive generation applies momentum-update encoders to generate diverse positive samples and reduce semantics corruption caused by graph augmentation. Then, the negative generation defines semantic integrity to determine which generated negative samples are more useful for positive samples, and the proposed negative selection dynamically selects negative samples of usefulness. In addition, the improved contrastive loss help selected negative samples to preserve their distinctive properties in sample space. Extensive experiments on molecular classification show the effectiveness of our MocGCL.

ACKNOWLEDGEMENT

This work was supported by The National Natural Science Foundation of China (No. 62076079), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

REFERENCES

[1] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," in *Journal of molecular biology vol. 330,4 (2003)*, 2003, pp. 771–83.

[2] P. Mohapatra, S. Chakravarty, and P. Dash, "Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system," vol. 28. Elsevier, 2016, pp. 144–160.

[3] M. Le Mercier, D. Hastir, X. Moles Lopez, N. De Neve, C. Maris, A.-L. Trepant, S. Rorive, C. Decaestecker, and I. Salmon, "A simplified approach for the molecular classification of glioblastomas." Public Library of Science San Francisco, USA, 2012.

[4] Y. Wang, Y. Min, X. Chen, and J. Wu, "Multi-view graph contrastive representation learning for drug-drug interaction prediction," in *Proceedings of the Web Conference 2021*, 2021, pp. 2921–2933.

[5] A. Olar and K. D. Aldape, "Using the molecular classification of glioblastoma to inform personalized treatment," vol. 232, no. 2. Wiley Online Library, 2014, pp. 165–177.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," vol. 286, no. 5439. American Association for the Advancement of Science, 1999, pp. 531–537.

[7] J. Tsang and G. M. Tse, "Molecular classification of breast cancer," vol. 27, no. 1. Wolters Kluwer, 2020, pp. 27–35.

[8] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," vol. 50, no. 5. ACS Publications, 2010, pp. 742–754.

[9] S. H. Alves, C. A. de Lanna, K. T. F. Leite, M. Boroni, and M. M. B. R. Vellasco, "Multi-omic data integration applied to molecular tumor classification," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 2171–2178.

[10] Y. Li, F.-X. Wu, and A. Ngom, "A review on machine learning principles for multi-view biological data integration," vol. 19, no. 2. Oxford University Press, 2018, pp. 325–340.

[11] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *In International Conference on Learning Representations*, 2018.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *arXiv preprint arXiv:1609.02907*, 2016.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017.

[14] C. Tan, J. Xia, L. Wu, and S. Z. Li, "Co-learning: Learning from noisy labels with self-supervision," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 1405–1413.

[15] J. Xia, C. Tan, L. Wu, Y. Xu, and S. Z. Li, "Ot cleaner: Label correction as optimal transport," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3953–3957.

[16] J. Xia, Y. Zhu, Y. Du, and S. Z. Li, "A survey of pretraining on graphs: Taxonomy, methods, and applications," in *arXiv preprint arXiv:2202.07893*, 2022.

[17] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 5812–5823.

[18] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 121–12 132.

[19] M. Sun, J. Xing, H. Wang, B. Chen, and J. Zhou, "Mocl: Contrastive learning on molecular graphs with multi-level domain knowledge," in *KDD 2021*, 2021.

[20] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." vol. 12, no. 9, 2011.

[21] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1365–1374.

[22] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and P. Yu, "Graph self-supervised learning: A survey." IEEE, 2022.

[23] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 824–10 832.

[24] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "Simgrace: A simple framework for graph contrastive learning without data augmentation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1070–1079.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[27] M. J. Zaki, W. Meira Jr, and W. Meira, "Data mining and analysis: fundamental concepts and algorithms." Cambridge University Press, 2014.

[28] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *International conference on machine learning*. PMLR, 2018, pp. 531–540.

[29] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," vol. 330, no. 4. Elsevier, 2003, pp. 771–783.

[30] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," vol. 21, no. suppl_1. Oxford University Press, 2005, pp. i47–i56.

[31] N. Wale, I. A. Watson, and G. Karypis, "Comparison of descriptor spaces for chemical compound retrieval and classification," vol. 14, no. 3. Springer, 2008, pp. 347–375.

[32] A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity," vol. 34, no. 2. ACS Publications, 1991, pp. 786–797.

[33] J. Kazius, R. McGuire, and R. Bursi, "Derivation and validation of toxicophores for mutagenicity prediction," vol. 48, no. 1. ACS Publications, 2005, pp. 312–320.

[34] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092.

[35] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 723–731.

[36] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," 2019.

[37] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He, "Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proceedings of the Web Conference 2021*, 2021, pp. 2081–2091.

[38] H. Tang, X. Liang, Y. Guo, X. Zheng, and B. Wu, "Graph fine-grained contrastive representation learning," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3478–3482.