# MoCL: Data-driven Molecular Fingerprint via Knowledge-aware Contrastive Learning from Molecular Graph

Mengying Sun
sunmeng2@msu.edu
Michigan State University
East Lansing, Michigan, USA

Jing Xing
xingjin1@msu.edu
Michigan State University
Grand Rapids, Michigan, USA

Huijun Wang
huijun.wang@agios.com
Agios Pharmaceuticals
Cambridge, Massachusetts, USA

Bin Chen
chenbi12@msu.edu
Michigan State University
Grand Rapids, Michigan, USA

Jiayu Zhou
jiayuz@msu.edu
Michigan State University
East Lansing, Michigan, USA

## ABSTRACT

Recent years have seen a rapid growth of utilizing graph neural networks (GNNs) in the biomedical domain for tackling drug-related problems. However, like any other deep architectures, GNNs are data hungry. While requiring labels in real world is often expensive, pretraining GNNs in an unsupervised manner has been actively explored. Among them, graph contrastive learning, by maximizing the mutual information between paired graph augmentations, has been shown to be effective on various downstream tasks. However, the current graph contrastive learning framework has two limitations. First, the augmentations are designed for general graphs and thus may not be suitable or powerful enough for certain domains. Second, the contrastive scheme only learns representations that are invariant to local perturbations and thus does not consider the global structure of the dataset, which may also be useful for downstream tasks. In this paper, we study graph contrastive learning designed specifically for the biomedical domain, where molecular graphs are present. We propose a novel framework called MoCL, which utilizes domain knowledge at both local- and global-level to assist representation learning. The local-level domain knowledge guides the augmentation process such that variation is introduced without changing graph semantics. The global-level knowledge encodes the similarity information between graphs in the entire dataset and helps to learn representations with richer semantics. The entire model is learned through a double contrast objective. We evaluate MoCL on various molecular datasets under both linear and semi-supervised settings and results show that MoCL achieves state-of-the-art performance.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**;
• **Applied computing** → **Bioinformatics**.


*KDD '21, August 14–18, 2021, Virtual Event, Singapore*

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467186

## KEYWORDS

Contrastive Learning, Molecular Graph, Domain knowledge

**ACM Reference Format:**
Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. 2021. MoCL: Data-driven Molecular Fingerprint via Knowledge-aware Contrastive Learning from Molecular Graph. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3447548.3467186

## 1 INTRODUCTION

Graph neural networks (GNNs) has been demonstrated to achieve state-of-the-art performance on graph-related tasks such as node classification [13, 38, 40], link prediction [48] and graph classification [9, 38, 42]. It has also been frequently used in the biomedical domain recently to tackle drug-related problems [18, 30, 32]. However, like most deep learning architectures, it requires large amount of labeled data to train whereas task-specific labels in real world are often of limited size (e.g., in biomedical domain, requiring labels such as drug responses from biological experiments is always expensive and time consuming). Therefore, pretraining schemes on GNNs have been actively explored recently.

One line of works focuses on designing pretext tasks to learn node or graph representations without labels. The predefined tasks include graph reconstruction [12, 14, 45] and context prediction [11, 22]. The other line follows a contrastive learning framework from computer vision domain [5, 41], in which two augmentations are generated for each data and then fed into an encoder and a projection head. By maximizing the mutual information between the two augmented views, the model is able to learn representations that are invariant to transformations. In particular, [44] proposed four types of augmentations for general graphs and demonstrated that contrastive learning on graphs is able to produce representations that are beneficial for downstream tasks.

However, contrastive learning on graphs has its unique challenges. First, the structural information and semantics of the graphs varies significantly across domains (e.g., social network v.s. molecular graphs), thus it is difficult to design universal augmentation scheme that fits all scenarios. It has been shown that general augmentations can be harmful under a specific domain context [44]. Second, most current graph contrastive learning frameworks learn invariant representations while neglect the global structure of the

entire data [1], e.g., some graphs should be closer in the embedding space due to their structural similarity. Nevertheless, modeling similarity between graphs itself is still a difficult problem [2]. Third, the contrast schemes are not unique because graph tasks can happen at different levels, e.g., node-graph contrast [10], node-node contrast [47], graph-graph contrast [44] are all possible contrast schemes.

Besides these unique challenges for graphs, contrastive learning itself also has unsolved problems. For example, accurately estimating mutual information in high dimension is difficult [23]. The connection between mutual information maximization and the success of contrastive learning is still not clear. In fact, [37] found the connection is actually weak, while instead metric learning shares some intrinsic connections with contrastive learning. These findings also motivate us to pay more attention to the role of augmentation schemes and global semantics of the data in order to improve contrastive learning on graphs.

Therefore, in this paper, we aim to tackle the aforementioned challenges in the context of biomedical domain, where molecular graphs are present. Our goal is to improve representations by infusing domain knowledge into the augmentation and constrast schemes. We propose to leverage both local-level and global-level domain knowledge to assist contrastive learning on molecular graphs. In particular, unlike general augmentations in which nodes and edges in a graph are randomly perturbed, we propose a new augmentation scheme called substructure substitution where a valid substructure in a molecule is replaced by a bioisostere that introduces variation without altering the molecular properties too much. The substitution rules are derived from domain resource and we regard it as local-level domain knowledge. The global-level domain knowledge encodes the global similarities between graphs. We propose to utilize such information to learn richer representations via a double contrast objective.

Leveraging domain knowledge to assist contrastive learning has rarely been explored in literature and our work is the first to make this attempt. In summary, our contributions are as follows:

- We propose a new augmentation scheme for molecular graphs based on local-level domain knowledge such that the semantics of graphs do not change in the augmentation process.
- We propose to encode global structure of the data into graph representations by adding a global contrast loss utilizing the similarity information between molecular graphs.
- We provide theoretical justifications that the learning objective is connected with triplet loss in metric learning which shed light on the effectiveness of the entire framework.
- We evaluate MoCL on various molecular datasets under both linear and semi-supervised settings and demonstrate its superiority over the state-of-the-art methods.

## 2 RELATED WORK

**Self-supervised learning on graphs.** A common strategy for learning node (graph) representation in an unsupervised manner is to design pretext tasks on unlabled data. For node-level tasks, You et al. [45] proposed three types of self-supervised tasks: node clustering, graph partition and graph completion to learn node representations. Peng et al. [22] proposed to predict the contextual position of a node relative to the other to encode the global topology into node representations. GPT-GNN [12] designed generative task in which node attributes and edges are alternatively generated such that the likelihood of a graph is maximized. After that, the pretrained GNN can be used for any downstream tasks. For graph level tasks, Hu et al. [11] first designed two tasks, predicting neighborhood context and node attributes to learn meaningful node representations, then using graph-level multi-task pretraining to refine the graph representation. GROVER [28] incorporated GNN into a Transformer-style architecture and learned node embedding by predicting contextual property and motif labels. Other works [31, 35, 43] utilized similar strategies for either node or graph level pretraining in the context of a more specific task or domain.

**Contrastive learning on graphs.** Contrastive learning on graphs can be categorized into two groups. One group aims to encode structure information by contrasting local and global representations. For example, DGI [39] proposed to maximize the mutual information between node embedding and graph summary vector to learn node representations that capture the graph semantics. InfoGraph [34] extended DGI to learn graph-level representations and further proposed a variant for semi-supervised scenarios. Another group aims to learn representations that are invariant to transformations, following the idea of contrastive learning on visual representations [5, 7, 41], where two augmentations (views) of an image are generated and fed into an encoder and a projection head, after which their mutual information is maximized. Similarly, You et al. [44] explored four types of augmentations for general graphs and demonstrated that the learned representations can help downstream tasks. Instead of general corruption, [10] used graph diffusion to generate the second view and performed contrast between node and graph from two views. GCA [47] proposed adaptive augmentation such that only unimportant nodes and edges are perturbed. However, GCA is focused on network data and not suitable for molecular graphs. Instead of focusing on augmentation views, MICRO-Graph [46] proposed to contrast based on sub-graphs (motifs). GCC [24] proposed to use random walk to generate subgraphs and contrast between them.

**Evaluation protocols.** There exist various evaluation schemes for graph level self-supervised learning. Most prior works [11, 34, 44, 46] adopt the linear evaluation protocol where a linear classifier is trained on top of the representations. [34, 44, 46] also adopt the semi-supervised protocol where only a small fraction of labels are available for downstream tasks. Other works [11, 28, 44] also explore the transfer learning setting in which the pretrained model is applied to other datasets.

## 3 METHOD

### 3.1 Problem Definition

A (molecular) graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, .., v_{|V|}\}$ and $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ denotes node and edge set respectively. Let $\mathbf{X} \in \mathbb{R}^{|V| \times d_1}$ be the feature matrix for all nodes in a graph, $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$ the adjacency matrix and $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times d_2}$ the edge features, our goal is to learn a graph encoder $\mathbf{h} = f(\mathbf{X}, \mathbf{A}, \mathbf{E}) \in \mathbb{R}^{d'}$ which maps an input graph to a vector representation without the presence of any labels. The learned encoder and representations can be used for downstream tasks directly or via finetune.

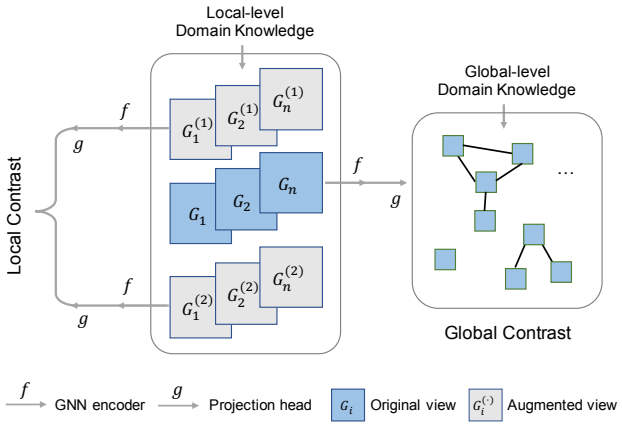



**Figure 1: Overall framework of MoCL. First, two augmented views are generated from local-level domain knowledge. Then, together with the original view (blue), they are fed into the GNN encoder and projection head. The local-level contrast maximizes the mutual information (MI) between two augmented views. The global-level contrast maximizes the MI between two similar graphs, where the similarity information is derived from global-level domain knowledge.**

## 3.2 Contrastive Learning Framework

In a conventional contrastive learning framework (Fig. 1 left), for each graph $G_i$, two augmentation operators $t_1$ and $t_2$ are sampled from the family of all operators $\mathcal{T}$, and applied to $G_i$ to obtain two correlated views $G_i^1 = t_1(G_i)$ and $G_i^2 = t_2(G_i)$. We use numbers in the **superscript** to represent different **views** throughout the paper. The correlated views are fed into a graph encoder $f$, producing graph representations $\mathbf{h}_i^1$ and $\mathbf{h}_i^2$, which are then mapped into an embedding space by a projection head $g$, yielding $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$. The goal is to maximize the mutual information between the two correlated views in the embedding space via Eq (1).

$$\mathcal{L}^{\text{local}} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i^{\text{local}}, \tag{1}$$

and the loss for each sample $\mathcal{L}_i^{\text{local}}$ can be written as:

$$\begin{aligned}\mathcal{L}_i^{\text{local}} &= \mathcal{L}_i^1 + \mathcal{L}_i^2 \\ &= -\log \underbrace{\frac{e^{s(\mathbf{z}_i^1,\mathbf{z}_i^2)/\tau}}{\sum_{j=1,j\neq i}^{n} e^{s(\mathbf{z}_i^1,\mathbf{z}_j^2)/\tau}}}_{\text{view 1 contrasts view 2}} - \log \underbrace{\frac{e^{s(\mathbf{z}_i^2,\mathbf{z}_i^1)/\tau}}{\sum_{j=1,j\neq i}^{n} e^{s(\mathbf{z}_i^2,\mathbf{z}_j^1)/\tau}}}_{\text{view 2 contrasts view 1}},\end{aligned} \tag{2}$$

where $n$ is the batch size, $s(\cdot,\cdot)$ is a function which measures the similarity of the two embeddings, $\tau$ is a scale parameter. The two correlated views $\mathbf{z}_i^1$ and $\mathbf{z}_i^2$ are regarded as positive pair while the rest pairs in the batch are regarded as negative pairs. The objective aims to increase the probability of occurrences of positive pairs as opposed to negative ones. Note that the negative pairs can be formed in two directions. If $\mathbf{z}_i^1$ is the anchor, all $\mathbf{z}_j^2$ in view 2 are contrasted; if $\mathbf{z}_i^2$ is the anchor, all $\mathbf{z}_j^1$ in view 1 are contrasted. Thus the loss for each sample consists of two parts as showed in Eq (2).

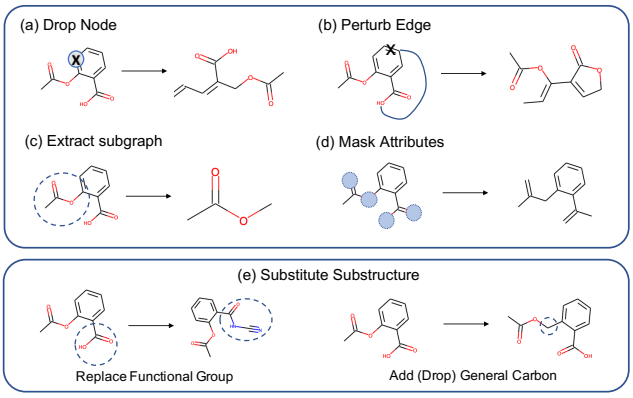



**Figure 2: Augmentation comparison. Upper: conventional augmentations that may alter the graph semantics. Lower: proposed augmentation in which valid substructures are replaced by bioisosteres that share similar properties.**

## 3.3 Local-level Domain Knowledge

Most existing approaches adopt random corruption during augmentation. For example, [47] proposed four types of augmentations for general graphs (Fig. 2 upper). However, such random corruption may alter the semantics of molecular graphs. For node dropping and edge perturbation, the resulting molecule is rarely biologically proper, e.g., dropping a carbon atom in the phenyl ring of aspirin breaks the aromatic system and results in an alkene chain (Fig. 2a); perturbing the connection of aspirin might introduce a five-membered lactone (Fig. 2b), which may drastically change the molecular properties. For subgraph extraction, the resulting structure is arbitrary and not representative for molecular functionality, e.g., methyl acetate is a sub group of aspirin (Fig. 2c), but also frequently shown in other compounds such as digitoxin and vitamin C with diverse chemical structures and biological effects. Enforcing high mutual information between such augmentation pairs may produce suboptimal representations for downstream tasks. This phenomenon has also been observed in [47] that edge perturbation deteriorates the performance of certain molecular tasks. Among the general augmentations, only attribute masking (Fig. 2d) does not violate the biological assumptions since it does not change the molecule, it only masks part of the atom and edge attributes.

Therefore, we aim to infuse domain knowledge to assist the augmentation process. We propose a new augmentation operator called *substructure substitution*, in which a valid substructure in a molecule is replaced by a bioisostere [17] which produces a new molecule with similar physical or chemical properties as the original one (Fig. 2e). We compile 218 such rules from domain resource [1]. Each rule consists of a source substructure and a target substructure represented by SMARTS string [2]. A sample rule is as follows:

```
[#6:2][#6:1](=O)[O;-,H1] >> [*:2][c:1]1nn[nH]n1
```

indicating the transition from left substructure (carboxylic acid) to the right one (nitrogen heterocycle). The substitution rules have

[1] https://www.schrodinger.com/drug-discovery
[2] https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html

| Group | # source | # target | Formula |
|---|---|---|---|
| CA | 1 | 68 | RCOO |
| Ester | 1 | 7 | RCOOR' |
| Ketone | 1 | 15 | ROR' |
| Phenyl | 22 | 36 | Aromatic Rings |
| Tbutyl | 1 | 10 | C4 |
| dsAmide | 4 | 18 | RONR'R" |
| msAmide | 2 | 32 | RONR' |
| nsAmide | 4 | 32 | RON |
| Total | 36 | 218 | - |

**Table 1: Source and target statistics for substitution rules. R/R'/R" represent arbitrary carbon-containing groups.**

36 unique source substructures which can be categorized into 8 groups. We summarize the statistics of the rules in Table 1. Note that target substructures are all unique and different. The original 218 substitution rules mostly happen at molecular positions where heteroatoms (heavy atoms that are not C or H) and aromatic rings are presented, therefore the variation for general carbon groups is limited. Under the common assumption that changing a few general carbon atoms will not alter the molecular property too much, we add 12 additional rules to subtract and add general carbon groups from and to a molecule. Some sample rules are:

```
[*:1][CH2][CH2][*:2] >> [*:1][*:2]  (drop)
[*:1]-[*:2] >> [*:1]CC[*:2]         (add)
```

Thus, MoCL consists of 230 rules in total to generate molecule variants that share similar properties. All the rules and code are available at https://github.com/illidanlab/MoCL-DK.

Moreover, since the source substructures in the rules are very common, a molecule may contain multiple source substructures or multiple copies of the same substructure in the rule, the proposed augmentation can be applied multiple times to generate variants with much more diversity. A notable difference between proposed augmentation and general augmentation is that the proposed rules are not guaranteed to be applicable to a molecule after it changes, therefore when applying proposed augmentation multiple times, we need to update the rule availability accordingly at each round. We summary the proposed augmentation procedure in Alg. 1.

## 3.4 Global-level Domain Knowledge

Maximizing mutual information between correlated views learns transformation-invariant representations. However, it may neglect the global semantics of the data. For example, some graphs should be closer in the embedding space since they share similar graph structures or semantics from domain knowledge. For molecular graphs, such information can be derived from multiple sources. For general graph structure, extended connectivity fingerprints (ECFPs) [27] encode the presence of substructures for molecules and are widely used to measure the structural similarity between molecular graphs. Drug-target networks [25] record the drug-protein interaction information which is one of the most informative biological activity measures. In this section, we first define graph similarity from general molecular graphs, then we propose two ways to incorporate the global semantics into our learning framework.

**Algorithm 1:** Pseudocode of domain augmentation.

**Input:** Molecule graph $G$, repeat time $R$, rules $\mathcal{T}$
**Output:** Augmented graph $G'$

```
for r = 1 to R do
    while T do
        sample t ~ T                # one augmentation rule
        {G^1, G^2, .., G^k} = t(G)  # all possible products
        random choose G = G^i
        update available T # rules may no longer be valid
        break;
G' = G
return G'
```

*3.4.1 Similarity calculation.* Given the ECFP of two molecules, $e_1, e_2 \in \{0,1\}^m$ where $m$ is the vector length and 1 indicates the presence of certain substructures, the similarity of $e_1$ and $e_2$ can be calculated as the Tanimoto coefficient [3]:

$$s(e_1, e_2) = \frac{N_{12}}{N_1 + N_2 - N_{12}}, \tag{3}$$

where $N_1, N_2$ denotes the number of 1s in $e_1, e_2$ respectively, and $N_{12}$ denotes the number of 1s in the intersection of $e_1, e_2$. The resulted coefficient $s(e_1, e_2) \in [0,1]$ and a larger value indicates higher structural similarity. Similarly, for drug-target network, $e_1, e_2 \in \{0,1\}^m$ becomes the interaction profile of a drug to all proteins where $m$ is the total number of proteins. The drug similarity can be calculated the same as Eq. (3).

*3.4.2 Global-level Objective.* We propose two strategies for using the global similarity information. One strategy is to use it as direct supervision. Given embeddings of two original graphs $\mathbf{z}_i$ and $\mathbf{z}_j$, we measure the similarity between them as $\theta(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^T \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$. We optimize the similarity using least square loss as follows:

$$\mathcal{L}_i^{\text{global}} = \sum\nolimits_{j \neq i} \mathcal{L}_{ij}^{\text{global}} = \sum\nolimits_{j \neq i} \|\theta(\mathbf{z}_i, \mathbf{z}_j) - s_{i,j}\|_2^2,$$

where $s_{i,j}$ is the similarity from Eq. (3).

The second strategy is to utilize a contrastive objective in which similar graph pairs have higher mutual information as compared to the background. The objective is written as:

$$\mathcal{L}_i^{\text{global}} = -\log \frac{\sum_{j=1, j \in \mathcal{N}_i}^{n} e^{s(\mathbf{z}_i, \mathbf{z}_j)/\tau}}{\sum_{j=1, j \notin \mathcal{N}_i}^{n} e^{s(\mathbf{z}_i, \mathbf{z}_j)/\tau}},$$

where $\mathcal{N}_i$ refers the neighbors of graph $i$. The neighbors can be derived from global similarity by setting a threshold or a neighborhood size. The global loss for all graphs thus becomes:

$$\mathcal{L}^{\text{global}} = \frac{1}{n} \sum\nolimits_{i=1}^{n} \mathcal{L}_i^{\text{global}}. \tag{4}$$

Finally, the full objective of the proposed MoCL can be written as:

$$\mathcal{L} = \mathcal{L}^{\text{local}} + \lambda \mathcal{L}^{\text{global}}, \tag{5}$$

where $\lambda$ is a tuning parameter that controls the emphasis between local loss and global loss. We summarize the pseudo code of the entire framework in Alg. 2.

**Algorithm 2:** Pseudocode of proposed framework.

**Input:** Molecule graphs $G$, rules $\mathcal{T}$, hyper parameter $\lambda$, number of epochs $M$
**Output:** Graph encoder $f$

```
for m = 1 to M do
    for iter = 1 to max_iter do
        G^1 = Alg.1(G, T), G^2 = Alg.1(G, T)
        h^1 = f(G^1), h^2 = f(G^2), h = f(G)
        z^1 = g(h^1), z^2 = g(h^2), z = g(h)
        Calculate local loss by Eq. (1)
        Calculate global loss by Eq. (4)
        Optimize f and g using Eq. (5)
return f
```

### 3.5 Connection to Metric Learning

It has been well studied that optimizing objective Eq. (1) is equivalent to maximizing a lower bound of the mutual information between the correlated views, also a lower bound of the mutual information between input and the hidden representations [6, 20]. Formally, denote $\mathbf{Z}^1$ and $\mathbf{Z}^2$ as the random variables for the embeddings of augmentations, $\mathbf{X}$ the variable for original input features:

$$\mathcal{L}^{\text{local}} \leq I(\mathbf{Z}^1; \mathbf{Z}^2) \leq I(\mathbf{X}; \mathbf{Z}^1, \mathbf{Z}^2).$$

Beyond mutual information maximization, in this section, we provide additional justification for the proposed method from the perspective of metric learning, which unifies the local and global objectives. We show the following important result:

Lemma 1. *Assume the projection head $g$ is an identity mapping, i.e., $\mathbf{z} = g(\mathbf{h}) = \mathbf{h}$, and the similarity function $s(\cdot,\cdot)$ is inner product, i.e., $s(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j$. Consider 1-nearest neighbor of each graph in the batch for global structure information, and $\lambda = 1$, the objective $\mathcal{L}_i$ is equivalent to the following:*

$$\mathcal{L}_i \propto \sum_{j \neq i} \underbrace{\|\mathbf{z}_i^1 - \mathbf{z}_i^2\|^2 - \|\mathbf{z}_i^1 - \mathbf{z}_j^2\|^2}_{\text{local contrast view 1}} + \underbrace{\|\mathbf{z}_i^2 - \mathbf{z}_i^1\|^2 - \|\mathbf{z}_i^2 - \mathbf{z}_j^1\|^2}_{\text{local contrast view 2}} + \sum_{j \neq k, k \in \mathcal{N}_i} \underbrace{\|\mathbf{z}_i - \mathbf{z}_k\|^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2}_{\text{global contrast}} + Const.$$

The lemma above connects the objective design to the metric learning. The equation consists of three triplet losses [4] which corresponds to the two local losses and the global loss respectively. As such, the MoCL objective aims to pull close the positive pairs while pushing away the negative pairs from both local and global perspective. Detailed proofs can be found in Appendix.

## 4 EXPERIMENT

In this section, we conduct extensive experiments to demonstrate the proposed method by answering the following questions:

**Q1**. Does local-level domain knowledge (MoCL-DK) learns better representations than general augmentations? How does combination of different augmentations behave?

| Dataset | # Tasks | Size | Avg. Node | Avg. Degree |
|---|---|---|---|---|
| bace | 1 | 1513 | 34.1 | 36.9 |
| bbbp | 1 | 2050 | 23.9 | 25.8 |
| clintox | 2 | 1483 | 26.1 | 27.8 |
| mutag | 1 | 188 | 17.8 | 19.6 |
| sider | 27 | 1427 | 33.6 | 35.4 |
| tox21 | 12 | 7831 | 18.6 | 19.3 |
| toxcast | 617 | 8597 | 18.7 | 19.2 |

**Table 2: Basic statistics for all datasets**

**Q2**. Does global-level domain knowledge (MoCL-DK-G) further improve the learned representations? Do the two proposed global losses perform the same?

**Q3**. How do the hyper-parameters ($\lambda$, neighbor size) involved in MoCL affect the model performance?

### 4.1 Evaluation Protocols

The evaluation process follows two steps. We first pretrain a model based on any comparison method, and then evaluate the learned model on downstream tasks. We adopt two evaluation protocols:

- Linear protocol: fix the representation from pretrained model and finetune a linear classifier on top of it.
- Semi-supervised protocol: sample a small set of labels of the downstream task and use the weights of learned graph encoder as initialization meanwhile finetune all the layers.

which are most commonly used in literature [11, 34, 44, 47].

### 4.2 Experimental Setup

**Datasets and Features.** We use 7 benchmark molecular datasets in the literature [11, 34, 44] to perform the experiments, which covers a wide range of molecular tasks such as binding affinity, response in bioassays, toxicity and adverse reactions:

- bace [33]: a dataset containing the binding results between molecules and human proteins .
- bbbp [16]: a dataset measuring the blood-brain barrier penetration property of molecules.
- mutag [29]: a dataset recording the mutagenic effect of a molecule on a specific gram negative bacterium.
- clintox & tox21 & toxcast [8, 19, 26]: datasets that contains the molecule toxicity from FDA clinical trials (clintox) and in vitro high-throughput screening (tox21 and toxcast).
- sider [15]: a dataset containing the adverse drug reactions (ADR) of FDA approved drugs.

The basic statistics of the datasets (size, tasks, molecule statistics) are summarized in Table 2. In this paper, we mainly focus on classification tasks as prior works [11, 34, 44], therefore we use AUC [36] as the major evaluation metric.

For molecular graphs, we use both atom features and bond features as inputs. We use i) atomic number and ii) chirality tag as features for atoms and i) bond type and ii) bond directions as features for chemical bonds [11].

**Model Architectures.** We use GIN [42] as our graph encoder $f$ which has been shown to be the most expressive graph neural network layer in prior works [11]. It also allows us to incorporate edge features of molecules into the learning process. The update rule for each GIN layer can be written as:

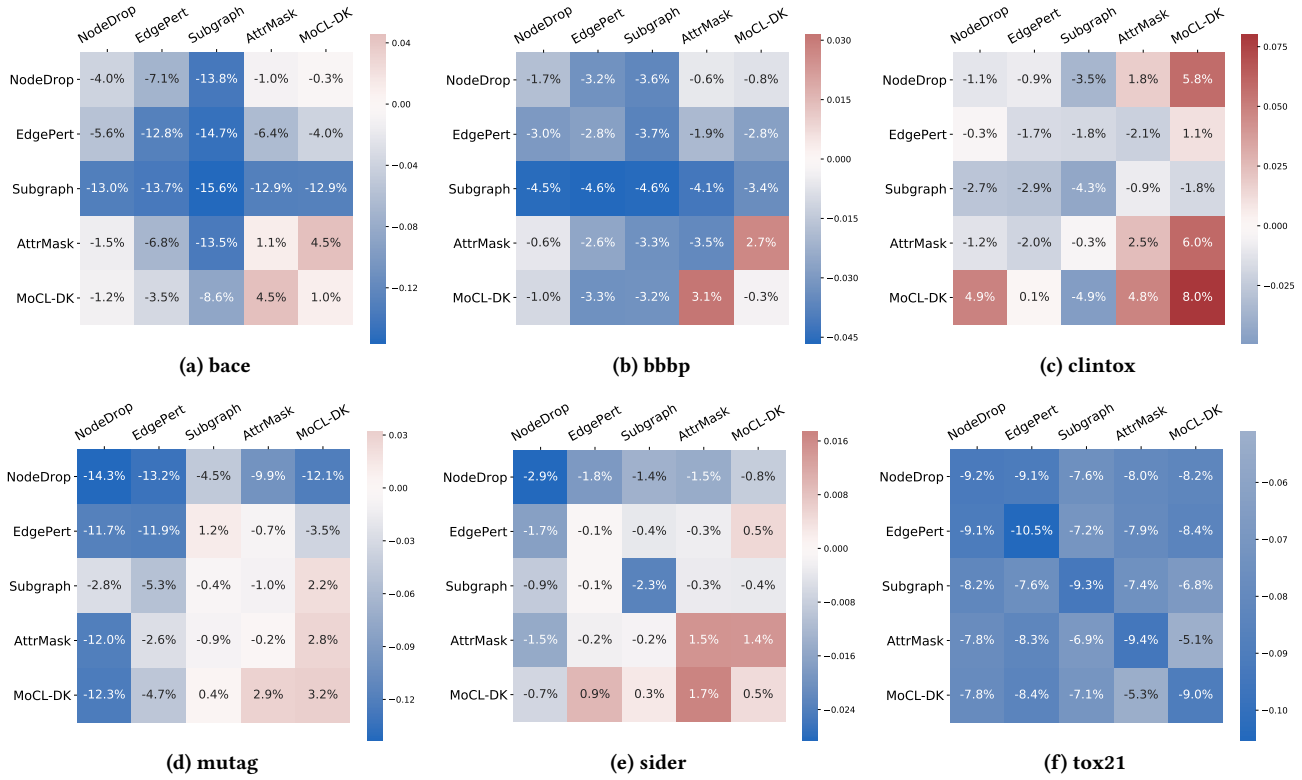



**Figure 3: Augmentation combination under linear evaluation protocol. Each cell represents the performance difference between i) a vanilla GNN trained from scratch (upper-bound) and ii) learned representations (fixed) from the pretrained model plus a linear classifier, under a given augmentation combination. Each number is averaged from 5 runs. Blue represents negative value and red positive. Higher value is better. MoCL-DK is the proposed augmentation with local-level domain knowledge.**

$$\mathbf{x}_i^{l+1} = \text{MLP}_\theta\left(\mathbf{x}_i^l + \sum\nolimits_{j\in\mathcal{N}_i} \text{ReLU}\left(\mathbf{x}_j^l + \mathbf{e}_{j,i}\right)\right),$$

where $\mathbf{x}_i^l$ is the node representation at $l$-th layer, $\mathcal{N}_i$ denotes the neighbor nodes of $i$-th node and $\mathbf{e}_{j,i}$ represents the edge feature between node $i$ and $j$. $\text{MLP}_\theta$ is a two-layer perceptron parameterized by $\theta$. Note that MLP here is for a single GIN layer in order to make the GIN layer the most expressive. After obtaining the node representations for all atoms in a molecule, we average them to get the graph representation $\mathbf{h}$.

We use another two-layer perceptron for the projection head $g$ in our framework following literature [5, 44]. It has been shown that a projection head with nonlinear transformation is necessary for a better representation of the layer before it due to information loss in the contrastive learning loss [5]. After adding a projection head, the representations at previous layer, ie., $\mathbf{h}$, can benefit more for downstream tasks. We use cosine similarity for the critic function $s(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j / \|\mathbf{z}_i\| \|\mathbf{z}_j\|$ [44].

**Baselines.** For both linear and semi-supervised evaluation protocols, we adopt three types of baselines for comparison:

- Vanilla GNN (Scratch): train a standard nonlinear GNN model on labeled data of the downstream task.
- General GNN self-supervised learning or pretraining baselines: i) InfoGraph [34], which maximizes the mutual information between nodes and graph; ii) Edge Pred & Context Pred [11]: which uses the node embeddings to predict graph edge and neighbor context in order to learn meaningful node representations; iii) Masking [11]: which masks the atom attributes and tries to predict them.
- Graph contrastive learning baselines: we adopt the four types of general augmentations for graph in [44]: i) node dropping; ii) edge perturbation; iii) subgraph extraction; iv) attribute masking for comparison. We also add linear procotol resutls reported in MICRO-Graph [46] which is a motif-based contrastive method for comparison (no public code available).

**Implementation Details.** We use 3 layers of GIN for all methods since 3-hops neighborhood covers most aromatic rings and is usually sufficient for molecular structure learning [27]. The dimensions for GIN layer and embedding layer are 512 and 128 respectively. We use Adam as optimizer with initial learning rate of 0.001 for all methods. We use dropout ratio 0.5 for GIN layers and default settings for baselines. The batch size is 32 across all scenarios. For pretraining models, the running epoch is fixed to 100. For downstream tasks, we use early stop via validation set. We implement all models using Pytorch [21] and run them on Tesla K80 GPUs.

The variation of results for a dataset comes from two sources, the pretrained model and the downstream task. By comparing them, we find the variation of pretrained model (by applying different seeds) is much smaller than the variation of downstream task (by

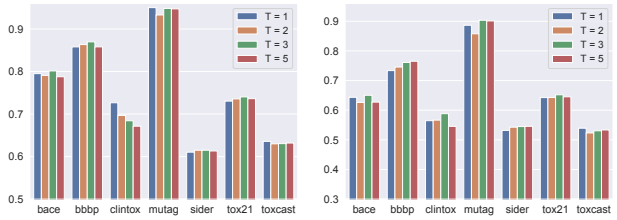
| Protocol | Linear Protocol | | | | | | | Semi-supervised Protocol | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method \| Dataset | bace | bbbp | clintox | mutag | sider | tox21 | toxcast | bace | bbbp | clintox | mutag | sider | tox21 | toxcast |
| scratch | 0.785 | 0.861 | 0.647 | 0.918 | 0.606 | 0.820 | 0.710 | 0.525 | 0.695 | 0.494 | 0.803 | 0.552 | 0.670 | 0.530 |
| InfoGraph | 0.594 | 0.611 | 0.458 | 0.771 | 0.502 | 0.615 | 0.562 | 0.614 | 0.735 | 0.487 | 0.887 | 0.523 | 0.589 | 0.535 |
| contextpred | 0.522 | 0.724 | 0.506 | 0.819 | 0.498 | 0.554 | 0.542 | 0.566 | 0.731 | 0.502 | 0.846 | 0.525 | 0.659 | 0.514 |
| edgepred | 0.662 | 0.592 | 0.504 | 0.622 | 0.502 | 0.500 | 0.501 | 0.604 | 0.694 | 0.486 | 0.915 | 0.545 | 0.615 | 0.529 |
| masking | 0.678 | 0.764 | 0.581 | 0.826 | 0.566 | 0.722 | 0.617 | 0.621 | **0.776** | 0.585 | 0.879 | **0.551** | 0.640 | 0.538 |
| drop_node | 0.746 | 0.843 | 0.635 | 0.775 | 0.577 | 0.728 | 0.633 | 0.603 | 0.767 | 0.492 | 0.836 | 0.542 | 0.656 | 0.525 |
| perturb_edge | 0.657 | 0.833 | 0.630 | 0.799 | 0.605 | 0.715 | 0.619 | 0.527 | 0.748 | 0.516 | **0.938** | 0.547 | 0.629 | 0.516 |
| subgraph | 0.629 | 0.815 | 0.603 | 0.914 | 0.583 | 0.727 | 0.625 | 0.565 | 0.769 | 0.539 | 0.918 | 0.548 | 0.656 | 0.514 |
| mask_attributes | 0.796 | 0.826 | 0.671 | 0.916 | 0.621 | 0.726 | 0.623 | 0.622 | 0.710 | 0.478 | 0.897 | 0.549 | **0.666** | 0.543 |
| MICRO-Graph | 0.708 | 0.830 | **0.735** | - | 0.573 | 0.718 | 0.595 | - | - | - | - | - | - | - |
| MoCL-DK | 0.801 | 0.870 | 0.727 | **0.950** | 0.615 | 0.740 | 0.636 | **0.650** | 0.765 | **0.588** | 0.903 | 0.546 | 0.645 | **0.539** |
| MoCL+AttrMask | **0.831** | **0.892** | 0.695 | 0.947 | **0.623** | **0.768** | **0.653** | 0.630 | 0.748 | 0.549 | 0.909 | 0.536 | 0.661 | 0.536 |
| MoCL-DK-G(LS) | 0.831 | 0.892 | 0.724 | 0.958 | 0.623 | **0.777**$^*$ | **0.659**$^*$ | 0.662 | 0.766 | 0.623 | 0.907 | 0.558 | 0.666 | **0.547**$^*$ |
| MoCL-DK-G(CL) | **0.845**$^*$ | 0.905 | **0.750**$^*$ | **0.969**$^*$ | **0.628**$^*$ | 0.768 | 0.653 | **0.706**$^*$ | **0.809**$^*$ | **0.623**$^*$ | 0.916 | 0.565 | 0.686 | 0.546 |
| MoCL+AttrMask-G(CL) | 0.833 | **0.911**$^*$ | 0.747 | 0.962 | 0.625 | 0.774 | 0.654 | 0.695 | 0.806 | 0.618 | 0.913 | **0.567**$^*$ | **0.687**$^*$ | 0.544 |

**Table 3: Averaged test AUC of comparison methods under linear and semi-supervised protocol (5 runs). Bold number denotes the best performance for local-level (augmentation) comparison. Bold$^*$ number denotes the best performance after incorporating global similarity information (MoCL-G). LS and CL represents least-square and contrastive global loss, respectively.**

different training-testing splits). Therefore, for each dataset, we use its molecular graphs to pretrain a model (1 seed) and then apply it to downstream task on the same dataset using different splits (5 seeds). We do not evaluate transfer learning setting in this paper where a pretrained model is applied to another dataset. During downstream task, we split the dataset into training (0.8), validation (0.1) and testing (0.1) set, we use validation set for early stop and evaluate the AUC on testing set. For semi-supervised protocol where only a small fraction of labels is used to train, since the data sizes are different, the ratio is picked from {0.01, 0.05, 0.5} such that around 100 molecules being selected for each dataset. For local-level domain knowledge, we use augmentation ratio 0.2 for general augmentations as prior work [44] and different augmentation times $\{1, 2, 3, 5\}$ for the proposed method. For example, MoCL-DK3 denotes applying domain augmentation 3 times. For global-level domain knowledge part, we try $\lambda = \{0.5, 1.0, 5.0, 10.0\}$ and 4 different nearest neighbor sizes for each dataset based on its size. We use ECFP with dimension 1024 to calculate the global similarity. The complete implementation details can be found in Appendix.

## 4.3 Local-level domain knowledge (Q1)

We first examine whether the proposed augmentation helps learn a better representation. Since the contrastive framework involves two correlated views, different augmentation schemes can be applied to each view. Figure 3 shows the results of different augmentation combinations under linear protocol for all datasets (the results of toxcast is similar as tox21 therefore we remove it due to space limit). MoCL-DK represent applying domain augmentation by only once. We can see that i) the representations from MoCL-DK (diagonals) plus a linear classifier yield prediction accuracies which are on-par with a deep learning model train from scratch (bace, bbbp, sider), or even better than it (clintox, mutag). ii) the proposed augmentation MoCL-DK combined with other augmentations almost always produce better results compared to other combinations (rows and columns that contain MoCL-DK are usually higher). iii) Attribute masking and MoCL-DK are generally effective across all scenarios,



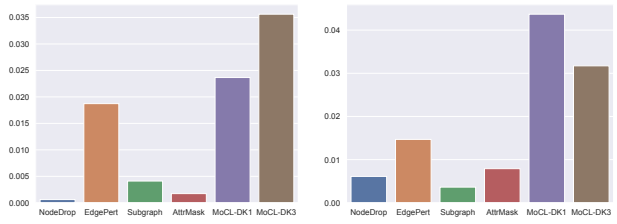

**(a) Linear protocol** **(b) Semi-supervised protocol**

**Figure 4: Average test AUC of MoCL-Local across different augmentation strengths (repeat times) for all datasets.**

combining them often yields even better performance. This verifies our previous assumption that MoCL-DK and attribute masking does not violate the biological assumption and thus works better than other augmentations. Moreover, harder contrast, e.g., combination of different augmentation schemes benefits more as compared to one augmentation schemes (MoCL-DK + AttrMask often produce the best results). This phenomenon is reasonable and also observed in prior works [44].

For semi-supervised protocol, the results are weaker, we did not include the augmentation combination figure due to space limit. But the complete results for all comparison methods for both linear and semi-supervised protocol can be found in Table 3, where the next-to-bottom panel represents results for proposed augmentation and the bottom panel presents global results which we will mention in the next subsection.

The proposed augmentation MoCL-DK can be applied multiple times to generate more complicated views. We tried over a range of different augmentation strengths and report the corresponding results for all datasets in Figure 4. We can see that for most datasets, as we apply more times the proposed augmentation, the performance first increases and then decreases. MoCL-DK3 usually achieves better results than others. For certain datasets (clintox, toxcast) the trend is not very clear between the two evaluation protocols.




(a) Linear protocol (b) Semi-supervised protocol

**Figure 5: Average test AUC gain from global domain knowledge for different augmentations across all datasets.**

| Protocol | Linear | | Semi-supervised | |
|---|---|---|---|---|
| Dataset | LS | CL | LS | CL |
| bace | 0.831 | **0.845** | 0.662 | **0.701** |
| bbbp | 0.891 | **0.903** | 0.766 | **0.809** |
| clintox | 0.724 | **0.750** | 0.608 | **0.619** |
| mutag | 0.954 | **0.963** | 0.895 | **0.907** |
| clintox | 0.623 | **0.628** | 0.551 | **0.563** |
| tox21 | 0.774 | 0.768 | 0.655 | **0.686** |
| toxcast | 0.659 | 0.653 | 0.547 | 0.546 |

**Table 4: Comparison between different global losses under MoCL-DK1 augmentation. LS: directly using global similarity and optimize by least-square loss; CL: contrastive loss using nearest neighbor derived from global similarity.**

## 4.4 Global-level domain knowledge (Q2)

We next study the role of global-level domain knowledge by examining the following sub-questions: i) Does global similarity helps general (baseline) augmentations? Does it helps the proposed augmentation? Are the effectiveness the same? ii) How do different global losses behave, i.e., direct supervision as least square loss v.s. contrastive loss, across all datasets, which one is better?

Figure 5 shows the performance gain by incorporating global similarity information for general (baseline) augmentations and the proposed augmentation. Each bar represents the median gain across all 7 datasets for a particular augmentation scheme. We can see that global information generally improves all augmentation schemes (the bars are positive). Interestingly, the gain for proposed domain augmentation (MoCL-DK1 and MoCL-DK3) are much higher as compared to other augmentations schemes. Note that we used the same set of global-level hyper-parameters for all augmentations for fair comparison. Table 4 shows the performance for different global losses under both evaluation protocols. We can see that contrastive loss (CL) for the global similarity achieves better results than directly using it as supervision by least-square loss (LS).

We summarize the complete results for all comparison methods in Table 3. We can see that i) contrastive learning works generally better than traditional graph pretraining methods, especially in linear protocol; ii) The proposed augmentation outperforms general augmentations. By combining MoCL augmentation and attribute masking, the results are even better for some datasets; iii) The global similarity information further improves the learned representations. Moreover, without combining with attribute masking, MoCL augmentation only already achieves the best performance under most scenarios after adding global information. The learned representations plus a linear classifier can achieve higher accuracy than a well-trained deep learning model. In summary, the proposed method is demonstrated to be effective for various molecular tasks.

## 4.5 Sensitivity Analysis (Q3)

Finally we check the sensitivity of global-level hyper-parameters, ie., the neighbor size and $\lambda$ that controls the weight between local and global loss. Figure 6 shows the performance surface under different hyper-parameter combinations of the proposed method for bbbp dataset. We can see that a relatively smaller neighbor size (not too small) and larger weights (not too large) for the global loss leads to a best result. Other datasets also show the similar pattern.

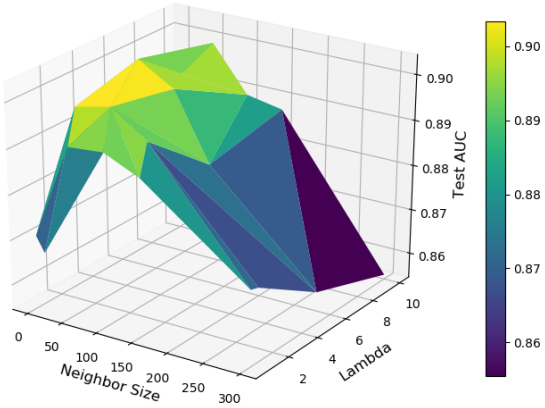



**Figure 6: Average test AUC of different neighbor size and $\lambda$ for MoCL-DK1-G under linear protocol (dataset: bbbp).**

## 4.6 Discussion

We provide additional observations and discussion in this subsection. First, we observe that representations which perform well under linear evaluation do not guarantee to be better in the semi-supervised setting. Since we finetune all the layers in semi-supervised learning, an overly delicate representation as initialization may not produce the best results in a fully nonlinear setting. Second, the effectiveness of contrastive learning also depends on the property of the dataset as well as the nature of the task. For example, single property prediction (mutag, bbbp) benefits more from pretraining as compared to toxicity prediction (tox21, toxcast) since it depends not only on the compound structure, but also the cellular environment. Therefore, incorporating drug-target network information and system biology data may be more helpful to these datasets, which is our future direction.

## 5 CONCLUSION

In this work, we propose to utilize multi-level domain knowledge to assist the contrastive representation learning on molecular graphs. The local-level domain knowledge enables new augmentation scheme and global-level domain knowledge incorporates global structure of the data into the learning process. We demonstrate that both knowledge improve the quality of the learned representations.

## Acknowledgment

This research is funded in part by National Science Foundation under grant IIS-1749940 (JZ), Office of Naval Research under grant N00014-20-1-2382 (JZ), National Institute of Health under grants 1R01GM134307 (JZ, BC) and K01ES028047 (BC).

## REFERENCES

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. 2019. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371* (2019).
[2] Yunsheng Bai, Hao Ding, Yang Qiao, Agustin Marinovic, Ken Gu, Ting Chen, Yizhou Sun, and Wei Wang. 2019. Unsupervised inductive whole-graph embedding by preserving graph proximity. (2019).
[3] Dávid Bajusz, Anita Rácz, and Károly Héberger. 2015. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics* 7, 1 (2015), 1–13.
[4] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2009. Large scale online learning of image similarity through ranking. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 11–14.
[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
[6] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.
[7] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks. Citeseer.
[8] Kaitlyn M Gayvert, Neel S Madhukar, and Olivier Elemento. 2016. A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology* 23, 10 (2016), 1294–1301.
[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*. PMLR, 1263–1272.
[10] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*. PMLR, 4116–4126.
[11] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* (2019).
[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1857–1867.
[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[14] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).
[15] Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. 2016. The SIDER database of drugs and side effects. *Nucleic acids research* 44, D1 (2016), D1075–D1079.
[16] Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. 2012. A Bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling* 52, 6 (2012), 1686–1697.
[17] Nicholas A Meanwell. 2011. Synopsis of some recent tactical application of bioisosteres in drug design. *Journal of medicinal chemistry* 54, 8 (2011), 2529–2591.
[18] Rocío Mercado, Tobias Rastemo, Edvard Lindelöf, Günter Klambauer, Ola Engkvist, Hongming Chen, and Esben Jannik Bjerrum. 2020. Graph Networks for Molecular Design. *Machine Learning: Science and Technology* (2020).
[19] Paul A Novick, Oscar F Ortiz, Jared Poelman, Amir Y Abdulhay, and Vijay S Pande. 2013. SWEETLEAD: an in silico database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery. *PloS one* 8, 11 (2013), e79568.
[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* (2019).
[22] Zhen Peng, Yixiang Dong, Minnan Luo, Xiao-Ming Wu, and Qinghua Zheng. 2020. Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604* (2020).
[23] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*. PMLR, 5171–5180.
[24] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.
[25] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. 2019. *Deep Learning for the Life Sciences*. O'Reilly Media. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.
[26] Ann M Richard, Richard S Judson, Keith A Houck, Christopher M Grulke, Patra Volarath, Inthirany Thillainadarajah, Chihae Yang, James Rathman, Matthew T Martin, John F Wambaugh, et al. 2016. ToxCast chemical landscape: paving the road to 21st century toxicology. *Chemical research in toxicology* 29, 8 (2016), 1225–1251.
[27] David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of chemical information and modeling* 50, 5 (2010), 742–754.
[28] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. 2020. Self-Supervised Graph Transformer on Large-Scale Molecular Data. *Advances in Neural Information Processing Systems* 33 (2020).
[29] Ryan Rossi and Nesreen Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.
[30] Miyuki Sakai, Kazuki Nagayasu, Norihiro Shibui, Chihiro Andoh, Kaito Takayama, Hisashi Shirakawa, and Shuji Kaneko. 2021. Prediction of pharmacological activities from chemical structures with graph convolutional neural networks. *Scientific reports* 11, 1 (2021), 1–14.
[31] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346* (2019).
[32] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. 2020. A deep learning approach to antibiotic discovery. *Cell* 180, 4 (2020), 688–702.
[33] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. 2016. Computational modeling of $\beta$-secretase 1 (BACE-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling* 56, 10 (2016), 1936–1949.
[34] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).
[35] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. 2020. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5892–5899.
[36] Mary M Tai. 1994. A mathematical model for the determination of total area under glucose tolerance and other metabolic curves. *Diabetes care* 17, 2 (1994), 152–154.
[37] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. 2019. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625* (2019).
[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[39] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax.. In *ICLR (Poster)*.
[40] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3733–3742.
[42] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).
[43] Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning*. PMLR, 10799–10808.
[44] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020).
[45] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2020. When does self-supervision help graph convolutional networks?. In *International Conference on Machine Learning*. PMLR, 10871–10880.
[46] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. 2020. Motif-Driven Contrastive Learning of Graph Representations. *arXiv preprint arXiv:2012.12533* (2020).
[47] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Graph Contrastive Learning with Adaptive Augmentation. *arXiv preprint arXiv:2010.14945* (2020).
[48] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (2018), i457–i466.

# APPENDIX

## Implementation Details

Table 5 shows the detailed parameter settings for all datasets. Semi-ratio depends on the data size such that around 100 molecule labels are sampled from each dataset. The neighbor size also depends on the data size such that the number of clusters is between 5 and 30 for all datasets. The parameter $\lambda$ which controls the weight between local and global loss, and augmentation time for MoCL-DK are all set to the same set of values for all datasets.

| Dataset | Size | Semi-ratio | Neigbor Size | $\lambda$ | DK |
|---|---|---|---|---|---|
| bace | 1513 | 0.05 | {50, 100, 150, 300} | {0.5, 1, 5, 10} | {1,2,3,5} |
| bbbp | 2050 | 0.05 | {50, 100, 150, 300} | {0.5, 1, 5, 10} | {1,2,3,5} |
| clintox | 1483 | 0.05 | {50, 100, 150, 300} | {0.5, 1, 5, 10} | {1,2,3,5} |
| mutag | 188 | 0.5 | {10, 20, 30, 40} | {0.5, 1, 5, 10} | {1,2,3,5} |
| sider | 1427 | 0.05 | {50, 100, 150, 300} | {0.5, 1, 5, 10} | {1,2,3,5} |
| tox21 | 7831 | 0.01 | (600, 800, 1000} | {0.5, 1, 5, 10} | {1,2,3,5} |
| toxcast | 8597 | 0.01 | {600, 800, 1000} | {0.5, 1, 5, 10} | {1,2,3,5} |

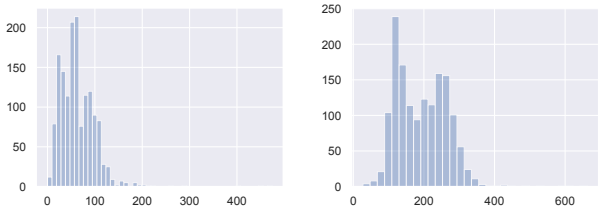
**Table 5: Detailed experimental settings for each dataset.**

Unlike prior work [44] in which only node, node features and connectivity information are used as input, our GNN incorporates edge features, therefore, the implementation of general augmentation is slightly different from [44]. We list the operations for both node (features) and edge (features) in Table 6.

| Augmentation | Node | Node features | Edge | Edge features |
|---|---|---|---|---|
| Drop Node | removed | removed | removed | removed |
| Perturb Edge | - | - | permuted | permuted |
| Subgraph | subsample | subsample | keep | keep |
| Mask Attributes | mask | mask | mask | mask |

**Table 6: Implementation details for general augmentation. Edge refers all edges that reach out from the corresponding node. - denotes no change.**

Figure 7 shows the distribution of number of augmentations that can be generated by applying MoCL-DK1 (left: from rules of substituting functional groups; right: from rules of adding/dropping general carbons). Other datasets reveal the same pattern therefore we do not include them due to space limit. We see that MoCL-DK1 can generate considerable number of augmentations for the molecules. If we apply MoCL-DK multiple times (MoCL-DK3, MoCL-DK5), the number of possible products can further increase drastically.




**(a) Function group rule** **(b) General carbon rule**

**Figure 7: Distribution of augmentations that can be generated by proposed augmentation rules (dataset: bace).**

## Proof of Lemma 1

Assume the projection head $g$ is an identity mapping, i.e., $\mathbf{z} = g(\mathbf{h}) = \mathbf{h}$, and the similarity function $s(\cdot,\cdot)$ is inner product, i.e., $s(\mathbf{z}_i, \mathbf{z}_j) = \mathbf{z}_i^T \mathbf{z}_j$. Consider 1-nearest neighbor of each graph in the batch for global structure information, and $\lambda = 1$, the objective $\mathcal{L}_i$ is equivalent to the following:

$$\mathcal{L}_i \propto \sum_{j\neq i} \underbrace{\|\mathbf{z}_i^1 - \mathbf{z}_i^2\|^2 - \|\mathbf{z}_i^1 - \mathbf{z}_j^2\|^2}_{\text{local contrast view 1}} + \underbrace{\|\mathbf{z}_i^2 - \mathbf{z}_i^1\|^2 - \|\mathbf{z}_i^2 - \mathbf{z}_j^1\|^2}_{\text{local contrast view 2}} + \sum_{j\neq k, k\in\mathcal{N}_i} \underbrace{\|\mathbf{z}_i - \mathbf{z}_k\|^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2}_{\text{global contrast}} + Const.$$

Proof.

$$\begin{aligned}
\mathcal{L}_i &= \log \frac{\sum_{j\neq i}^n e^{s(\mathbf{z}_i^1, \mathbf{z}_j^2)}}{e^{s(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau}} + \log \frac{\sum_{j\neq i}^n e^{s(\mathbf{z}_i^2, \mathbf{z}_j^1)/\tau}}{e^{s(\mathbf{z}_i^2, \mathbf{z}_i^1)/\tau}} \\
&\quad + \log \frac{\sum_{j\neq k, k\in\mathcal{N}_i}^n e^{s(\mathbf{z}_i, \mathbf{z}_j)/\tau}}{e^{s(\mathbf{z}_i, \mathbf{z}_k)}} \\
&= \log \sum_{j\neq i}^n e^{s(\mathbf{z}_i^1, \mathbf{z}_j^2)/\tau - s(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau} + \log \sum_{j\neq i}^n e^{s(\mathbf{z}_i^2, \mathbf{z}_j^1)/\tau - s(\mathbf{z}_i^2, \mathbf{z}_i^1)/\tau} \\
&\quad + \log \sum_{j\neq k, k\in\mathcal{N}_i}^n e^{s(\mathbf{z}_i, \mathbf{z}_j)/\tau - s(\mathbf{z}_i, \mathbf{z}_k)/\tau}
\end{aligned}$$

By applying first-order Taylor expansion we have:

$$\begin{aligned}
\mathcal{L}_i &\approx \sum_{j\neq i}^n e^{s(\mathbf{z}_i^1, \mathbf{z}_j^2)/\tau - s(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau} + \sum_{j\neq i}^n e^{s(\mathbf{z}_i^2, \mathbf{z}_j^1)/\tau - s(\mathbf{z}_i^2, \mathbf{z}_i^1)/\tau} \\
&\quad + \sum_{j\neq k, k\in\mathcal{N}_i}^n e^{s(\mathbf{z}_i, \mathbf{z}_j)/\tau - s(\mathbf{z}_i, \mathbf{z}_k)/\tau} - 3 \\
&\approx \frac{1}{\tau}\Big[ \sum_{j\neq i}^n s(\mathbf{z}_i^1, \mathbf{z}_j^2) - s(\mathbf{z}_i^1, \mathbf{z}_i^2) + \sum_{j\neq i}^n s(\mathbf{z}_i^2, \mathbf{z}_j^1) - s(\mathbf{z}_i^2, \mathbf{z}_i^1) \\
&\quad + \sum_{j\neq k, k\in\mathcal{N}_i}^n s(\mathbf{z}_i, \mathbf{z}_j) - s(\mathbf{z}_i, \mathbf{z}_k)\Big] - 3 \\
&= \frac{1}{\tau}\Big[ \sum_{j\neq i}^n \mathbf{z}_i^{1T} \mathbf{z}_j^2 - \mathbf{z}_i^{1T} \mathbf{z}_i^2 + \sum_{j\neq i}^n \mathbf{z}_i^{2T} \mathbf{z}_j^1 - \mathbf{z}_i^{2T} \mathbf{z}_i^1 \\
&\quad + \sum_{j\neq k, k\in\mathcal{N}_i}^n \mathbf{z}_i^T \mathbf{z}_j - \mathbf{z}_i^T \mathbf{z}_k\Big] - 3 \\
&= \frac{1}{2\tau}\Big[ \sum_{j\neq i}^n \|\mathbf{z}_i^1 - \mathbf{z}_i^2\|^2 - \|\mathbf{z}_i^1 - \mathbf{z}_j^2\|^2 + \|\mathbf{z}_i^2 - \mathbf{z}_i^1\|^2 - \|\mathbf{z}_i^2 - \mathbf{z}_j^1\|^2 \\
&\quad + \sum_{j\neq k, k\in\mathcal{N}_i}^n \|\mathbf{z}_i - \mathbf{z}_k\|^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2\Big] - 3 \\
&\propto \sum_{j\neq i} \|\mathbf{z}_i^1 - \mathbf{z}_i^2\|^2 - \|\mathbf{z}_i^1 - \mathbf{z}_j^2\|^2 + \|\mathbf{z}_i^2 - \mathbf{z}_i^1\|^2 - \|\mathbf{z}_i^2 - \mathbf{z}_j^1\|^2 \\
&\quad + \sum_{j\neq k, k\in\mathcal{N}_i} \|\mathbf{z}_i - \mathbf{z}_k\|^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2 - 6\tau
\end{aligned}$$

■