MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks

Yuyang Wang, Jianren Wang, Zhonglin Cao, Amir Barati Farimani[*]
Carnegie Mellon University
yuyangw@cmu.edu, jianrenwang.cs@gmail.com,
zhonglic@andrew.cmu.edu, barati@cmu.edu

Abstract

Molecular machine learning bears promise for efficient molecule property prediction and drug discovery. However, due to the limited labeled data and the giant chemical space, machine learning models trained via supervised learning perform poorly in generalization. This greatly limits the applications of machine learning methods for molecular design and discovery. In this work, we present *MolCLR*: Molecular Contrastive Learning of Representations via Graph Neural Networks (GNNs), a self-supervised learning framework for large unlabeled molecule datasets. Specifically, we first build a molecular graph, where each node represents an atom and each edge represents a chemical bond. A GNN is then used to encode the molecule graph. We propose three novel molecule graph augmentations: atom masking, bond deletion, and subgraph removal. A contrastive estimator is utilized to maximize the agreement of different graph augmentations from the same molecule. Experiments show that molecule representations learned by *MolCLR* can be transferred to multiple downstream molecular property prediction tasks. Our method thus achieves state-of-the-art performance on many challenging datasets. We also prove the efficiency of our proposed molecule graph augmentations on supervised molecular classification tasks.

1 Introduction

Molecular representation is fundamental and essential in design of functional and novel chemical compounds [1, 2, 3, 4]. Due to the enormous magnitude of possible stable chemical compounds, development of an informative representation to generalize among the entire chemical space can be challenging [5, 6, 7]. Conventional molecular representations, like SMILES [8] and ECFP [9], have became standard tools in computational chemistry. Recently with the development of machine learning methods, data-driven molecular representation learning and its applications, including chemical property prediction [10, 11, 12, 13, 14], chemical modeling [15, 16, 17], and drug discovery [18, 19, 20, 21, 22], have gathered growing attentions.

However, learning such representations can be difficult due to three major challenges. Firstly, it is hard to represent the molecular information thoroughly. For instance, string-based representations, like SMILES [8], SMARTS [23], and SELFIES [24], fail to encode the important topology information directly. To preserve the rich structural information, many recent works exploit Graph Neural Networks (GNNs) [25, 26], and have shown promising results in molecular property prediction [13, 27, 28] and virtual screening [29, 30]. Secondly, the magnitude of chemical space is enormous [31], e.g., the size of potential pharmacologically active molecules is estimated to be in the order of 10^{60} [32]. This places a great difficulty for any molecular representations to generalize among the potential chemical compounds. Thirdly, labeled data for molecular learning tasks are expensive and

[*] Corresponding author: barati@cmu.edu.

far from sufficient, especially when compared with the size of potential chemical space. Obtaining labels of molecular property usually requires sophisticated and time-consuming lab experiments [33]. The breadth of chemical research further complicates the challenges, since the properties of interest range from quantum mechanics to biophysics [34, 35]. Consequently, the number of labels in most molecular learning benchmarks is far from adequate. Machine learning models trained on such benchmarks can easily get over-fitting and perform poorly on molecules dissimilar to the training set.

In this work, we propose Molecular Contrastive Learning of Representations (MolCLR) via Graph Neural Networks to address all the above challenges. MolCLR is a self-supervised learning framework trained on the large unlabeled molecule dataset. Through contrastive loss [36, 37], MolCLR learns the representations by contrasting positive molecule graph pairs against negative ones. Three molecule graph augmentation strategies are introduced: atom masking, bond deletion, and subgraph removal. Molecule graph pairs augmented from the same molecule are denoted as positive, while others are denoted as negative. A widely-used GNN model, Graph Isomorphism Network (GIN) [26], is pre-trained through MolCLR to extract informative representation from the augmented molecule graph. The pre-trained model is then fine-tuned on the downstream molecular property prediction tasks. Experiments show that the performance of our MolCLR surpasses other self-supervised learning and pre-training strategies in multiple molecular benchmarks [34]. Besides, in the downstream tasks, our MolCLR rivals or even exceeds supervised learning baselines, which include sophisticated graph convolution operations for molecules or domain-specific featurization. We also demonstrate that our molecule graph augmentation strategies improve the performance of supervised learning on molecular benchmarks when utilized as a direct data augmentation plug-in.

To summarize, (1) We propose MolCLR, a self-supervised learning framework for molecular representation learning. (2) We propose three molecule graph augmentation strategies to generate contrastive pairs, namely atom masking, bond deletion, and subgraph removal. Besides, we also demonstrate the improvement of implementing our proposed molecule graph augmentations in supervised molecular classifications. (3) We achieve the state-of-the-arts on several downstream molecular classification tasks with fine-tuning. This indicates that the MolCLR is capable of learning informative molecular representations without domain knowledge.

2 Preliminaries

Contrastive Learning. Contrastive learning [38] aims at learning representation through contrasting positive data pairs against negative ones. In [39], CNN is trained by discriminating between surrogate classes parameterized by feature vectors. Memory bank is then introduced in [40] to store features of each instances. Several works have then adopted and improved the memory bank [41, 42]. MoCo [43, 44] proposes a moving-average momentum encoder, which builds an on-the-fly consistent dictionary. Instead of using memory bank, SimCLR [37, 45] demonstrates contrastive learning can greatly benefits from the composition of data augmentations and large batch sizes. Based on InfoNCE loss [36], SimCLR proposes the normalized temperature-scaled cross entropy (NT-Xent) loss as given in Eq. 1:

$$\mathcal{L}_{i,j} = \log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}\{k \neq i\} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \tag{1}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are latent vectors extracted from a positive data pair, N is the batch size, $\text{sim}(\cdot)$ measures the similarity between the two vectors, and τ is the temperature parameter.

Graph Neural Networks. Non-euclidean data represented in the form of graphs is common across various domains [46, 47], such as molecule structures we investigate in this work [48]. A graph G is defined as $G = (V, E)$, where V and E are nodes and edges respectively [49, 50]. Modern Graph Neural Networks (GNNs) utilize a neighborhood aggregation operation, which update the node representation iteratively [51, 52, 53, 25]. The aggregation update rule for a node feature on the k-th layer of a GNN is given in Eq. 2:

$$\boldsymbol{a}_v^{(k)} = \text{AGGREGATE}^{(k)}(\{\boldsymbol{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}), \; \boldsymbol{h}_v^{(k)} = \text{COMBINE}^{(k)}(\boldsymbol{h}_v^{(k-1)}, \boldsymbol{a}_v^{(k)}), \tag{2}$$

where $\boldsymbol{h}_v^{(k)}$ is the feature of node v at the k-th layer and $\boldsymbol{h}_v^{(0)}$ is initialized by node feature $\boldsymbol{x}_v$. $\mathcal{N}(v)$ denotes the set of all the neighbors of node v. To further extract a graph-level feature $\boldsymbol{h}_G$, readout operation integrates all the node features among the graph G as given in Eq. 3:

$$\boldsymbol{h}_G = \text{READOUT}(\{\boldsymbol{h}_u^{(k)} : v \in G\}). \tag{3}$$

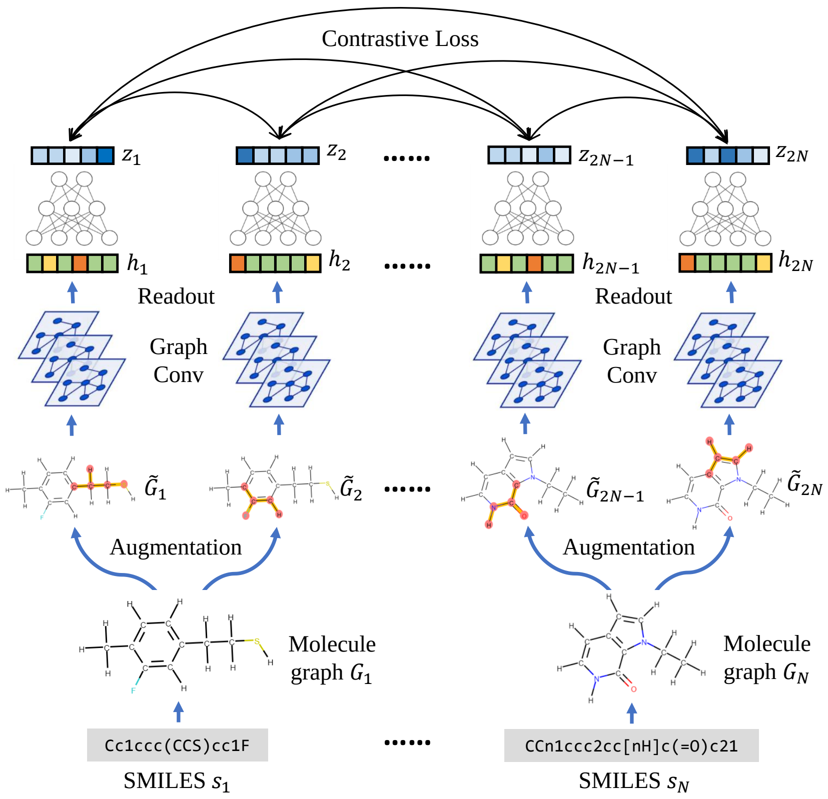


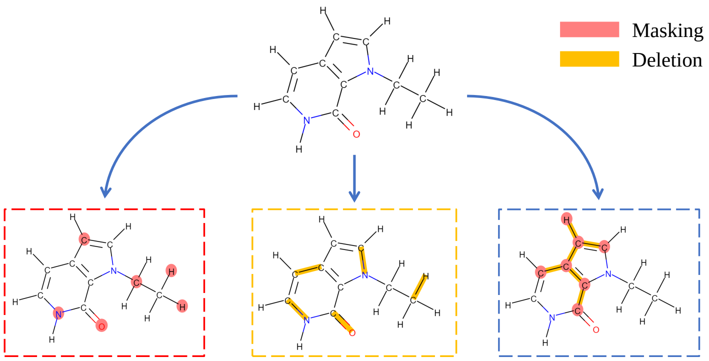
Figure 1: Molecular Contrastive Learning of Representations via Graph Neural Networks. A SMILES s_n from a mini-batch of N molecule data is converted to a molecule graph G_n. Two stochastic molecule graph data augmentation operators are applied to each graph, resulting two correlated masked graphs: $\tilde{G}_{2n-1}$ and $\tilde{G}_{2n}$. A base feature encoder built upon graph convolutions and the readout operation extracts the representation h_{2n-1}, h_{2n}. Contrastive loss is utilized to maximize agreement between the latent vectors z_{2n-1}, z_{2n} from the MLP projection head.

Various aggregation operations have been proposed to improve the performance of GNN. GraphSAGE [54] proposes a max-pooling operation over a ReLU [55] activated linear transformation as the aggregation. GCN [25] integrates the aggregation and combination operations by introducing a mean-pooling over the node itself and its adjacencies before the linear transformation. GIN [26] utilizes an MLP and weighted summation of node features in the aggregation. GAT [56] performs the multi-head attention to increase the model's expressive capability. Various other aggregation operations include diffusion convolution [57], message passing [13], and Gaussian-kernelized weight function [58]. Besides, incorporating edge features into the GNN has also been investigated [13, 59, 60, 61, 28, 62]. For readout operations, common strategies can be a graph-level summation, averaging, and max pooling. Some works have also developed elaborate graph readout modules to improve predictive performance and computational efficiency of GNNs [63, 64, 59, 65].

3 Method

3.1 MolCLR Framework

Our MolCLR model is developed upon the contrastive learning framework [37, 45]. Latent representations from positive augmented molecule graph pairs are contrasted with representations from negative pairs. As shown in Figure 1, the whole pipeline is composed of four components: data processing and augmentation, GNN-based feature extractor, non-linear projection head, and an NT-Xent [37] contrastive loss.



(a) Atom masking (b) Bond deletion (c) Subgraph removal

Figure 2: Three molecule graph augmentation strategies. (a) **Atom masking** randomly replaces the node feature $\boldsymbol{x}_v$ of an atom feature with a mask token $\boldsymbol{m}$. (b) **Bond deletion** randomly deletes the bond between two atoms, so that the they are not directly connected on the graph. (c) **Subgraph removal** randomly removes an induced subgraph [66] from the original molecule graph. Within the subgraph, all nodes are masked and all edges are deleted.

Given a SMILES data s_n from a mini-batch of size N, the corresponding molecule graph G_n is built, in which each node represents an atom and each edge represents a chemical bond between atoms. Using molecule graph augmentation strategies (explained in Section 3.2), G_n is transformed into two different but correlated molecule graphs: $\tilde{G}_i$ and $\tilde{G}_j$, where $i = 2n - 1$ and $j = 2n$. In this work, three molecule graph augmentation strategies, including atom masking, bond deletion, and subgraph removal, are applied in composition and stochastically. Molecule graphs augmented from the same molecule are denoted as positive pairs, whereas those from different molecules are denoted as negative pairs. The feature extractor $f(\cdot)$ maps the augmented molecule graphs into the representations $h_i, h_j \in \mathbb{R}^d$. Various GNNs can be plugged in to model $f(\cdot)$. In our case, we implement the commonly-used GIN [26] aggregation operation and an average pooling as the readout operator to extract the molecular representations. A non-linear projection head $g(\cdot)$ is modeled by an MLP with one hidden layer, which maps the representations h_i and h_j into latent vectors z_i and z_j respectively. Contrastive loss, NT-Xent, is applied to the $2N$ latent vectors z's as given in Eq. 1, and cosine similarity is utilized to calculate $\text{sim}(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|_2 \|z_j\|_2}$.

3.2 Molecule Graph Augmentation

We employ three molecule graph data augmentation strategies (Figure 2) as transformations for our MolCLR framework: atom masking, bond deletion, and subgraph removal.

Atom Masking Atoms in the molecule graph are randomly masked with a given ratio. When an atom is masked, its atom feature $\boldsymbol{x}_v$ is replaced by a mask token $\boldsymbol{m}$, which is distinguished from any atom features in the molecular graph. Shown by red shadows in Figure 2(a), six atoms including two Hydrogen, two Carbon, a Nitrogen, and an Oxygen are masked in the molecule graph.

Bond Deletion Bond deletion randomly deletes chemical bonds between the atoms with a certain ratio. Unlike atom masking which substitutes the original feature with a mask token, bond deletion is a more rigorous augmentation as it removes the edges completely from the molecule graph. Such a stronger augmentation strategy forces the GNN feature extractor to learn more informative representations.

Subgraph Removal Subgraph removal can be considered as a combination of atom masking and bond deletion. Subgraph removal starts from a randomly picked origin atom. The removal process proceeds by masking the neighbors of the original atom, and then the neighbors of the neighbors, until the number of masked atoms reaches a given ratio of the total number of atoms in the molecular graph. The bonds between the masked atoms are then deleted, such that the masked atoms and deleted bonds form an induced subgraph [66] of the original molecule graph. As shown in Figure 2(c), the removed subgraph includes all the bonds between the masked atoms.

4 Experiments

4.1 Datasets

Pre-training Dataset. For MolCLR pre-training, we use 10 million unique unlabeled molecule SMILES from [67], which are collected from PubChem [68]. RDKit [69] is then utilized to add hydrogen atoms to molecule structure and then build the molecule graphs from the SMILES strings. Within the molecule graph, each node represents an atom and each edge represents a chemical bond. We randomly split the pre-training dataset into training and validation set with a ratio of 95/5.

Downstream Datasets. To benchmark the performance of our MolCLR framework, we use 7 datasets from MoleculeNet [34], containing in total 44 binary classification tasks. These tasks cover molecule properties of multiple domains, including physical chemistry, biophysics, and physiology. For each dataset, we use the scaffold split [70] from DeepChem [71] to create an 80/10/10 train/valid/test split as suggested in [60]. Unlike the common random split, the scaffold split, which is based on molecular substructures, makes the prediction task more challenging yet realistic.

4.2 Baselines

Supervised learning models. We comprehensively evaluate the performance of our MolCLR model with supervised learning methods. For shallow machine learning models, Random Forest (RF) [72] and Support Vector Machine (SVM) [73] are implemented, which take molecular descriptors as the input. Besides, state-of-the-art graph-based neural networks are also included. Extended GIN [26, 60] with edge feature involved in aggregation is compared. D-MPNN [28] and MGCNN [74], which are graph neural network models designed specifically for molecule prediction tasks, are also included as the baselines.

Self-supervised learning models. To better demonstrate the power of our MolCLR framework, we further include other molecular self-supervised learning models in the baselines. HU. et.al [60] with both node-level and graph-level pre-training is considered. It should be pointed out that though node-level pre-training is based on self-supervision, the graph-level pre-training is supervised on some molecule property labels [60]. N-Gram graph [75] is also implemented, which computes a compact representation directly through the molecule graph.

4.3 Training Details

Each atom on the molecule graph is embedded by its atomic number and chirality type [76], while each bond is embedded by its type and direction. We implement a 5-layer Graph Isomorphism Network (GIN) [26] with ReLU activation [55] as the GNN backbone, and follow the modification in [60] to make GIN compatible with edge features. An average pooling is applied on each graph as the readout operation to extract the 512-dimension molecular representation. An MLP with one hidden layer maps the representation into a 256-dimension latent space. Adam [77] optimizer with weight decay 10^{-5} is used to optimize the NT-Xent loss. After the initial 10 epochs with learning rate 3×10^{-4}, cosine annealing without restart [78] decays the learning rate cyclically. The model is trained with batch size 512 for the total 100 epochs. The pre-training of MolCLR takes ~5 days on one NVIDIA Quadro RTX 6000.

For the downstream task fine-tuning, we add a randomly initialized 2-layer MLP with ReLU activation on top of the base feature extractor. For binary classification tasks, softmax cross-entropy loss is implemented. The learning rate of the MLP head is set to 3×10^{-4} and the base GIN extractor is set to 3×10^{-5}. The model is trained using Adam optimizer with batch size 32 for another 50 epochs. For each task, we fine-tune the pre-trained model three times using different random seeds for scaffold splitting to get the average and standard deviation of the performance. The whole framework is implemented based on Pytorch Geometric [79].

Table 1: Test ROC-AUC (%) performance comparison of different models, where the first five models are supervised learning methods and the last three are self-supervised/pre-training methods. Mean and standard deviation on each benchmark are reported.

Dataset	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV
# Molecules	2039	7831	1478	41127	1513	1478	93087
# Tasks	1	12	2	1	1	27	17
RF	71.4±0.0	76.9±1.5	71.3±5.6	78.1±0.6	**86.7±0.8**	**68.4±0.9**	63.2±2.3
SVM	72.9±0.0	**81.8±1.0**	66.9±9.2	**79.2±0.0**	86.2±0.0	**68.2±1.3**	67.3±1.3
MGCN [74]	**85.0±6.4**	70.7±1.6	63.4±4.2	73.8±1.6	73.4±3.0	55.2±1.8	70.2±3.4
D-MPNN [28]	71.2±3.8	68.9±1.3	**90.5±5.3**	75.0±2.1	85.3±5.3	63.2±2.3	**76.2±2.8**
HU. et.al [60]	70.8±1.5	78.7±0.4	78.9±2.4	80.2±0.9	85.9±0.8	65.2±0.9	81.4±2.0
N-Gram [75]	**91.2±3.0**	76.9±2.7	85.5±3.7	**83.0±1.3**	87.6±3.5	63.2±0.5	81.6±1.9
MolCLR	73.6±0.5	**79.8±0.7**	**93.2±1.7**	80.6±1.1	**89.0±0.3**	**68.0±1.1**	**88.6±2.2**

4.4 Results on Downstream Tasks

Table 1 demonstrates the test ROC-AUC performance of our MolCLR model in comparison to baseline models. The average and standard deviation of three individual runs are reported. Bold cells denote the best performing method on each benchmark, either via supervised or self-supervised/pre-training strategy. Observations from Table 1 are the followings. (1) In comparison with other self-supervised learning or pre-training strategies, our MolCLR framework achieves the best performance on 5 out of 7 benchmarks, with an average improvement of 5.0%. Such improvement illustrates that our MolCLR is a powerful self-supervised learning strategy, which is easy to implement and requires little domain-specific sophistication. (2) Compared with the supervised learning baselines, MolCLR also shows rival performance. In some benchmarks, our pre-training model even surpasses the SOTA supervised learning methods. For instance, on ClinTox, MolCLR improves the ROC-AUC by 2.9%. (3) Notably, MolCLR performs remarkably well on datasets with a limited number of molecules, like Clitox, BACE, and SIDER benchmarks. The performance validates that MolCLR learns informative representations that can be transferred among different datasets. Such capacity of generalization bears promise for predicting potential molecular properties in drug discovery and design.

4.5 Ablation Study

4.5.1 Temperature in Contrastive Loss

Table 2: Test ROC-AUC (%) performance comparison of different temperature parameter τ. Mean and standard deviation of all the seven benchmarks are reported.

Temperature (τ)	0.05	0.1	0.5
ROC-AUC (%)	76.8±1.2	80.2±1.3	78.4±1.7

The choice of the temperature parameter τ in Eq. 1 impacts the performance of contrastive learning [37]. An appropriate τ benefits the model to learn from hard negative samples. To investigate τ for molecule representation learning, we train MolCLR with three different temperatures: 0.05, 0.1, and 0.5 as shown in Table 2. We report the averaged ROC-AUC over all the seven benchmarks using 25% subgraph removal as the augmentation strategy. It is demonstrated that $\tau = 0.1$ performs the best in the downstream molecular tasks. Therefore, we use this temperature setting in the following experiments.

4.5.2 Composition of Molecule graph Augmentations

To systematically investigate the effect of molecule graph augmentation strategies, we compare different compositions of atom masking, bond deletion, and subgraph removal. Shown in Figure 3 are the ROC-AUC mean and standard deviation of each data augmentation compositions on different benchmarks. Four augmentation compositions are considered. (1) Integration of atom masking and bond deletion with both ratios p set to 25%. (2) Subgraph removal with a random ratio p from 0% to

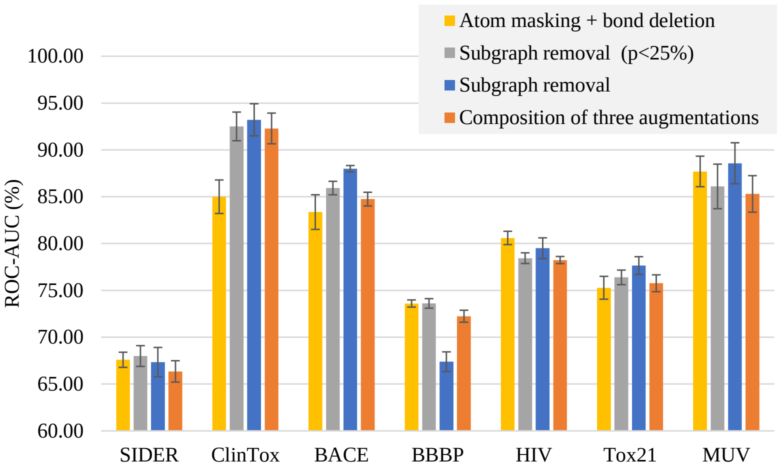


Figure 3: Test ROC-AUC (%) performance of pre-trained MolCLR model with different compositions of molecular graph augmentation strategies. Height of each bar represents the mean ROC-AUC on the benchmark, and length of each error bar represents the standard deviation.

25%. (3) Subgraph removal with a fixed 25% ratio. (4) Composition of all the three augmentation strategies. Specifically, a subgraph removal with a random ratio 0% to 25% is first applied. Then if the ratio of masked atoms is smaller than 25%, we continue random atom masking until it reaches the ratio of 25%. Similarly, if the bond deletion ratio is smaller than 25%, more bonds are deleted to reach the set ratio. The four compositions are shown in yellow, gray, blue, and orange respectively in Figure 3.

As Figure 3 illustrates, subgraph removal with a 25% ratio reaches the best performance on average among all the four compositions. This could because that subgraph removal is an intrinsic composition of atom masking and bond deletion, and that subgraph removal further disentangles the local substructures compared with strategy (1). However, subgraph removal with a fixed 25% ratio performs poorly in BBBP dataset, which can be attributed to that molecule structures in BBBP are sensitive, such that a slight topology change can cause great property difference. Besides, it is worth noticing that the composition of all three augmentations, (4), does not improve the performance. On the contrary, it hurts the ROC-AUC compared with single subgraph removal augmentation in most benchmarks. The composition of all the three augmentation strategies can remove a wide range of substructures within the molecule graph, thus eliminate the important topology information.

4.6 Molecule Graph Augmentation on Supervised Molecular Classifications

Table 3: Test ROC-AUC (%) of GIN with/without molecule graph augmentations on all the seven supervised molecular classification benchmarks. GIN models are trained in the supervised learning manner without pre-training.

Dataset	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV
GIN w/o Aug	65.8±4.5	74.0±0.8	58.0±4.4	75.3±1.9	70.1±5.4	57.3±1.6	71.8±2.5
GIN w/ Aug	72.1±0.9	75.0±1.1	64.0±2.4	76.1±1.2	71.6±0.7	65.2±1.4	80.5±3.1

The molecule graph augmentation strategies in our work, namely atom masking, bond deletion, and subgraph removal, can be implemented as a direct data augmentation plug-in for any graph-based molecular learning methods. To validate the effectiveness of molecule graph augmentations on supervised molecular tasks, we train GIN models with/without augmentations from scratch without pre-training. Specifically, subgraph masking with a fixed ratio 25% is implemented as the augmentation. Table 3 documents the mean and standard deviation of test ROC-AUC over the seven molecular property classification benchmarks. On all the seven benchmarks, GINs trained with augmentations surpass the models without augmentations, and improve the averaged ROC-AUC score

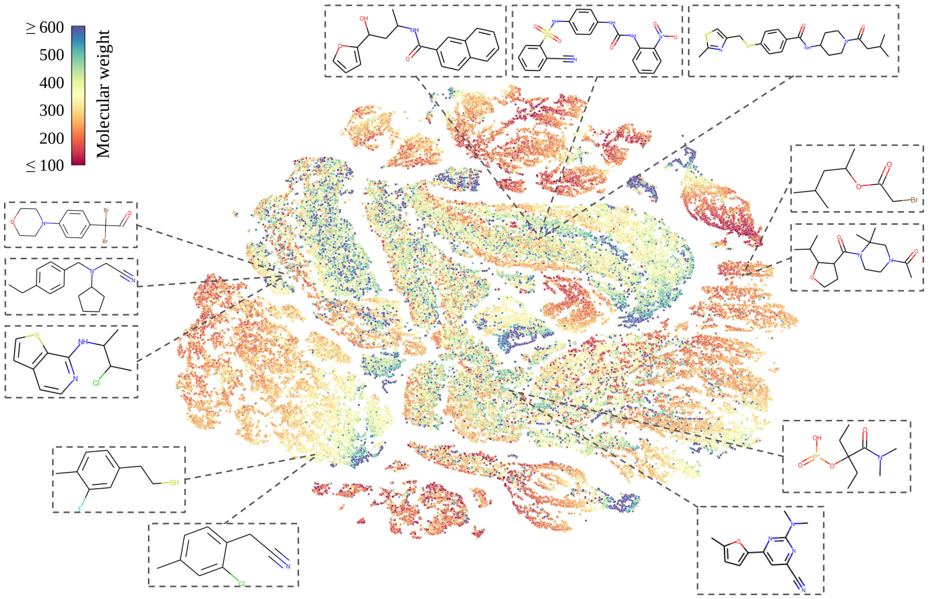


Figure 4: Two-dimensional t-SNE embedding of the molecular representations learned by our MolCLR pre-training. Representations are extracted from the validation set of the pre-training dataset, which contains 100k unique molecules. The color of each embedding point indicates its corresponding molecular weight.

by 7.2%. Implementation of our molecule graph augmentation strategies on supervised molecular property prediction tasks improves the performance greatly even without pre-training.

4.7 MolCLR Representation Visualization

We examine the representations learned by pre-trained MolCLR using t-SNE embedding [80]. The t-SNE algorithm maps similar molecular representations to adjacent points. Shown in Figure 4 are 100K molecules from validation set of the pre-training data embedded to 2D via t-SNE, colored based on the molecular weights. We also include some randomly selected molecules in the figure to illustrate what are the similar/dissimilar molecules learned by MolCLR pre-training. As shown in Figure 4, MolCLR learns close representations for molecules with similar topology structures and functional groups. For instance, the three molecules shown on the top possess carbonyl groups connected with aryls. The two molecules shown on the bottom left have similar structures, where a halogen atom (Fluorine or Chlorine) is connected to benzene. This demonstrates the potentials of MolCLR in application of drug search and drug discovery.

5 Related Works

Molecular representation learning has been growing rapidly over the last decade with the development and success of machine learning, especially deep learning driven by neural networks [81, 10, 20]. In conventional cheminformatics, molecules are represented in unique fingerprint vectors, such as Extended-Connectivity Fingerprints (ECFP) [9]. Given the fingerprints, deep neural networks are built to predict certain properties or classes [82, 83, 84]. Besides, SMILES [8], which maps the molecule into a string, is also widely-used for molecule representation [12, 85]. Language models built upon RNNs are direct fit for learning representation from SMILES [11, 86, 87, 88]. With the recent success of transformer-based architectures, such language models have been also utilized in molecular representation learning from SMILES strings [89, 90]. Recently, GNNs, which naturally encode the structure information, have been introduced to molecular representation learning [10, 48, 91]. MPNN [13] and D-MPNN [28] implement a message-passing architecture to aggregate the information from molecule graphs. Further, SchNet [27] models quantum interactions within molecules in the GNN. DimNet [61] integrates the directional information by transforming messages based on the angle between atoms.

Benefiting from the growth of available molecule data [92, 93, 34, 68], self-supervised/pre-trained molecular representation learning has also been investigated. Self-supervised language models, like BERT [94], has been implemented to learn molecular representation with SMILES as input [67, 95]. On molecule graph, N-Gram Graph [75] builds the representation for the graph by assembling the vertex embedding in short walks, which needs no training. HU. et.al [60] propose both node-level and graph-level tasks for GNN pre-training. However, the graph-level pre-training is based on supervised-learning tasks, which is constraint by limited labels. Our MolCLR framework, on the contrary, learns graph-level features directly via contrastive loss without any property labels.

6 Conclusions and Future Works

In this work, we investigate self-supervised learning for molecular representation. Specifically, we propose Molecular Contrastive Learning of Representations (MolCLR) via GNNs and three molecular graph augmentations strategies: atom masking, bond deletion, and subgraph removal. Through contrasting positive pairs against negative pairs from augmentations, MolCLR learns informative representation with general GNN backbones. Experiments show that MolCLR pre-trained GNN models achieve great improvement on various molecular benchmarks, and show better generalizations compared with models trained in the supervised learning manner.

Molecular representations learned by MolCLR demonstrates the transferability to molecular tasks with limited data and the power of generalization on the large chemical space. There exist many promising directions to investigate as future works. For instance, improvement of the GNN backbones (e.g. transformer-based GNN architectures [96]) can help extract better molecular representations. Besides, visualization and interpretation of self-supervised learned representations are of great interest [97]. Such investigations can help researchers better understand chemical compounds and benefit drug discovery.

References

[1] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18):184115, 2013.

[2] Luca M Ghiringhelli, Jan Vybiral, Sergey V Levchenko, Claudia Draxl, and Matthias Scheffler. Big data of materials science: critical role of the descriptor. *Physical review letters*, 114(10):105503, 2015.

[3] Bing Huang and O Anatole Von Lilienfeld. Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity, 2016.

[4] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):1–22, 2020.

[5] Tudor I Oprea and Johan Gottfries. Chemography: the art of navigating in chemical space. *Journal of combinatorial chemistry*, 3(2):157–166, 2001.

[6] Richard Bade, Ho-Fung Chan, and Jóhannes Reynisson. Characteristics of known drug space. natural products, their derivatives and synthetic drugs. *European journal of medicinal chemistry*, 45(12):5646–5652, 2010.

[7] Aaron M Virshup, Julia Contreras-García, Peter Wipf, Weitao Yang, and David N Beratan. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19):7296–7303, 2013.

[8] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[9] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[10] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, page 2224–2232, Cambridge, MA, USA, 2015. MIT Press.

[11] Stanisław Jastrzębski, Damian Leśniak, and Wojciech Marian Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

[12] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[14] Mohammadreza Karamad, Rishikesh Magar, Yuting Shi, Samira Siahrostami, Ian D Gates, and Amir Barati Farimani. Orbital graph convolutional neural network for material property prediction. *Physical Review Materials*, 4(9):093801, 2020.

[15] Stefan Chmiela, Huziel E Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications*, 9(1):1–10, 2018.

[16] Volker L Deringer, Noam Bernstein, Albert P Bartók, Matthew J Cliffe, Rachel N Kerber, Lauren E Marbella, Clare P Grey, Stephen R Elliott, and Gábor Csányi. Realistic atomistic structure of amorphous silicon from machine-learning-driven molecular dynamics. *The journal of physical chemistry letters*, 9(11):2879–2885, 2018.

[17] Wujie Wang and Rafael Gómez-Bombarelli. Coarse-graining auto-encoders for molecular dynamics. *npj Computational Materials*, 5(1):1–9, 2019.

[18] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

[19] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

[20] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019.

[21] Rishikesh Magar, Prakarsh Yadav, and Amir Barati Farimani. Potential neutralizing antibodies discovered for novel corona virus using machine learning. *arXiv preprint arXiv:2003.08447*, 2020.

[22] Cheng-Hao Liu, Maksym Korablyov, Stanisław Jastrzębski, Paweł Włodarczyk-Pruszyński, Yoshua Bengio, and Marwin HS Segler. Retrognn: Approximating retrosynthesis by graph neural networks for de novo drug design. *arXiv preprint arXiv:2011.13042*, 2020.

[23] Daylight Chemical Information Systems, Inc. `https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html`. (Accessed Feb 13 2021).

[24] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

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

[26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[27] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

[28] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[29] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[30] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS omega*, 4(14):15956–15965, 2019.

[31] Peter Kirkpatrick and Clare Ellis. Chemical space, 2004.

[32] Regine S Bohacek, Colin McMartin, and Wayne C Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1):3–50, 1996.

[33] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

[34] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[35] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11, 2020.

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[37] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[38] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[39] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. Citeseer, 2014.

[40] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[41] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[42] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[44] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[45] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[46] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[47] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

[48] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[49] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[50] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[51] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

[52] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[53] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

[54] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

[55] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

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

[57] James Atwood and Don Towsley. Diffusion-convolutional neural networks. *arXiv preprint arXiv:1511.02136*, 2015.

[58] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[59] Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9211–9219, 2019.

[60] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

[61] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[62] Yulei Yang and Dongsheng Li. Nenn: Incorporate node and edge features in graph neural networks. In *Asian Conference on Machine Learning*, pages 593–608. PMLR, 2020.

[63] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[64] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804*, 2018.

[65] Nicolò Navarin, Dinh Van Tran, and Alessandro Sperduti. Universal readout for graph convolutional neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[66] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[67] Seyone Chithrananda, Gabe Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

[68] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019.

[69] Greg Landrum. Rdkit: Open-source cheminformatics, 2006.

[70] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[71] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

[72] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[73] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[74] Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1052–1060, 2019.

[75] S. Liu, M. F. Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In *NeurIPS*, 2019.

[76] Alan D McNaught, Andrew Wilkinson, et al. *Compendium of chemical terminology*, volume 1669. Blackwell Science Oxford, 1997.

[77] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[78] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[79] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[80] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[81] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[82] Thomas Unterthiner, Andreas Mayr, Günter Klambauer, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, and Sepp Hochreiter. Deep learning as an opportunity in virtual screening. In *Proceedings of the deep learning workshop at NIPS*, volume 27, pages 1–9, 2014.

[83] Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

[84] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[85] Anvita Gupta, Alex T Müller, Berend JH Huisman, Jens A Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for de novo drug design. *Molecular informatics*, 37(1-2):1700111, 2018.

[86] Zheng Xu, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*, pages 285–294, 2017.

[87] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[88] Francesca Grisoni, Michael Moret, Robin Lingwood, and Gisbert Schneider. Bidirectional molecule generation with recurrent neural networks. *Journal of chemical information and modeling*, 60(3):1175–1183, 2020.

[89] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

[90] Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzębski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

[91] Evan N. Feinberg, Debnil Sur, Zhenqin Wu, Brooke E. Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S. Pande. Potentialnet for molecular property prediction. *ACS Central Science*, 4(11):1520–1530, 2018.

[92] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[93] Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

[94] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[95] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pages 429–436, 2019.

[96] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[97] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.