nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-022-00447-x

Check for updates

Molecular contrastive learning of representations via graph neural networks

Yuyang Wang[1,2], Jianren Wang[3], Zhonglin Cao[1] and Amir Barati Farimani[1,2,4] ✉

Molecular machine learning bears promise for efficient molecular property prediction and drug discovery. However, labelled molecule data can be expensive and time consuming to acquire. Due to the limited labelled data, it is a great challenge for supervised-learning machine learning models to generalize to the giant chemical space. Here we present MolCLR (Molecular Contrastive Learning of Representations via Graph Neural Networks), a self-supervised learning framework that leverages large unlabelled data (~10 million unique molecules). In MolCLR pre-training, we build molecule graphs and develop graph-neural-network encoders to learn differentiable representations. Three molecule graph augmentations are proposed: atom masking, bond deletion and subgraph removal. A contrastive estimator maximizes the agreement of augmentations from the same molecule while minimizing the agreement of different molecules. Experiments show that our contrastive learning framework significantly improves the performance of graph-neural-network encoders on various molecular property benchmarks including both classification and regression tasks. Benefiting from pre-training on the large unlabelled database, MolCLR even achieves state of the art on several challenging benchmarks after fine-tuning. In addition, further investigations demonstrate that MolCLR learns to embed molecules into representations that can distinguish chemically reasonable molecular similarities.

Molecular representation is fundamental and essential in the design of functional and novel chemical compounds[1–3]. 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[4]. Conventional molecular representations, such as extended-connectivity fingerprints (ECFP)[5], have become 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[6–8], chemical modelling[9–11] and molecule design[12–14], have gathered growing attention.

However, learning such representations can be difficult due to three major challenges. First, it is hard to represent the molecular information thoroughly. For instance, string-based representations, like SMILES[15] and SELFIES[16], fail to encode the important topology information directly. To preserve the rich structural information, many recent works exploit graph neural networks (GNNs)[17,18], and have shown promising results in molecular property prediction[7,19,20]. Second, the magnitude of chemical space is enormous[21], for example, the size of potential pharmacologically active molecules is estimated to be in the order of 10^{60} (ref. [22]). This places a great difficulty for any molecular representations to generalize among the potential chemical compounds. Third, labelled data for molecular learning tasks are expensive and far from sufficient, especially when compared with the size of potential chemical space. Obtaining labels of molecular properties usually requires sophisticated and time-consuming lab experiments[23]. The breadth of chemical research further complicates the challenges because the properties of interest range from quantum mechanics to biophysics[24]. Consequently, the number of labels in most molecular learning benchmarks is far from adequate. Machine learning models trained on such limited data can easily get over-fit and perform poorly on molecules dissimilar to the training set.

Molecular representation learning has been growing rapidly over the past decade with the development and success of machine learning, especially deep neural networks (DNNs)[6,25,26]. In conventional cheminformatics, molecules are represented in unique fingerprint (FP) vectors, such as ECFP. Given the FPs, DNNs are built to predict certain properties[27–29]. Besides the FP, string-based representations (like SMILES) are widely used for molecular learning[30,31]. Language models built on RNNs are a direct fit for learning representation from SMILES[32,33]. With the recent success of transformer-based architectures, such language models have been also utilized in learning representations from SMILES[34,35]. Recently, GNNs, which naturally encode structural information, have been introduced to molecular representation learning[6,36]. MPNN[7] and D-MPNN[20] implement a message-passing architecture to aggregate the information from molecule graphs. Further, SchNet[19] models quantum interactions within molecules in the GNN. DimeNet[37] integrates the directional information by transforming messages based on the angle between atoms.

Benefiting from the growth of available molecule data[24,38–40], self-supervised/pre-trained molecular representation learning has also been investigated. Self-supervised language models, like BERT[41], have been implemented to learn molecular representation with SMILES as input[42,43]. On molecule graph, N-Gram Graph[44] builds the representation for the graph by assembling the vertex embedding in short walks, which needs no training. Hu et al.[45] 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 constrained by limited labels. You et al.[46] extends the contrastive learning to unstructured graph data, but the framework is not specifically designed for molecule graph learning and is only trained on limited molecular data.

[1]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. [2]Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA. [3]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. [4]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. ✉e-mail: barati@cmu.edu

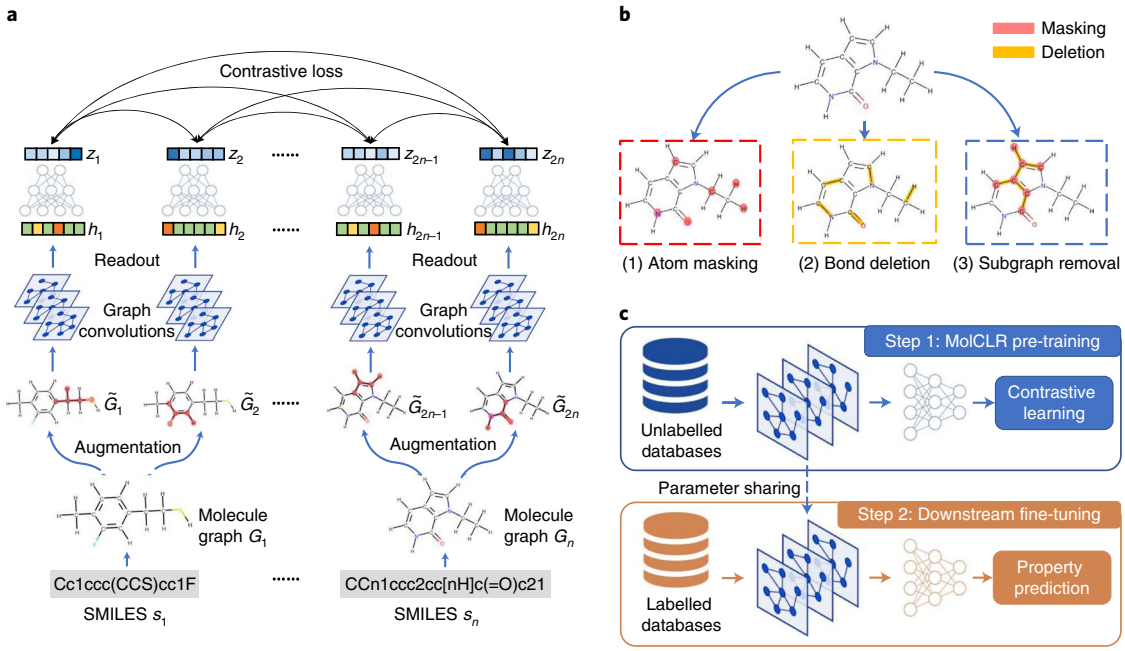

Fig. 1 | Overview of MolCLR. a, MolCLR pre-training. A SMILES s_n from a batch of N molecule data is converted to a molecule graph G_n. Two stochastic molecule graph data augmentation operators are applied to each graph, resulting two correlated masked graphs: $\tilde{G}_{2n-1}$ and $\tilde{G}_{2n}$. A base feature encoder built on 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. **b**, Molecule graph augmentation strategies: atom masking, bond deletion and subgraph removal. **c**, The whole MolCLR framework. GNNs are first pre-trained via MolCLR to learn representative features. Fine-tuning for downstream molecular property predictions shares the pre-trained parameters of the GNN encoder and randomly initializes an MLP head. It then follows the supervised learning to train the model.

In this Article, we propose MolCLR (Molecular Contrastive Learning of Representations via Graph Neural Networks) shown in Fig. 1 to address all of the above challenges. MolCLR is a self-supervised learning framework trained on the large unlabelled dataset with around 10 million unique molecules. Through contrastive loss[47,48], 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. Widely used GNN models, graph convolutional network (GCN)[17] and graph isomorphism network (GIN)[18], are developed as GNN encoders in MolCLR to extract informative representation from molecule graphs. The pre-trained model is then fine-tuned on the downstream molecular property prediction benchmarks from MoleculeNet[24]. In comparison to GCN and GIN trained via supervised learning, our MolCLR significantly improves the performance on both classification and regression tasks. Benefiting from pre-training on the large database, MolCLR surpasses other self-supervised learning and pre-training strategies in multiple molecular benchmarks. Moreover, on several tasks, our MolCLR rivals or even exceeds supervised learning baselines that 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. Further comparison between MolCLR representations and conventional FPs indicates that MolCLR learns to distinguish molecular similarities from pre-training on the large unlabelled data. Data and code for this work can be found in the CodeOcean capsule[49].

To summarize, (1) we propose MolCLR, a self-supervised learning framework for molecular representation learning; (2) three molecule graph augmentation strategies are introduced to generate contrastive pairs, namely atom masking, bond deletion and subgraph removal; (3) benefiting from pre-training on large unlabelled data, simple GNN models trained via MolCLR demonstrate significant improvements on all molecular benchmarks in comparison to supervised learning; (4) MolCLR even boosts simple GNN models to the state of the art (SOTA) on several molecular benchmarks with fine-tuning, compared to more sophisticated GNNs, which cannot utilize unlabelled data.

Results

MolCLR framework. Our MolCLR model is developed upon the contrastive learning framework[48,50]. Latent representations from positive augmented molecule graph pairs are contrasted with representations from negative pairs. The whole pipeline (Fig. 1a) is composed of four components: data processing and augmentation, GNN-based feature extractor, non-linear projection head, and the normalized temperature-scaled cross-entropy (NT-Xent) [48] contrastive loss.

Given a SMILES data s_n from a 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, G_n is transformed into two different but correlated molecule graphs: $\tilde{G}_i$ and $\tilde{G}_j$, where $i=2n-1$ and $j=2n$. Molecule graphs augmented from the same molecule are denoted as a positive pair, whereas those from different molecules are denoted as negative pairs. The feature extractor $f(\cdot)$ is modelled by GNNs and maps the molecule graphs into the representations $h_i, h_j \in \mathbb{R}^d$. In our case, we implement GCN[17] and GIN[18] with an average pooling as the feature extractor. A non-linear projection head $g(\cdot)$ is modelled by an MLP with one hidden layer, which maps

Table 1 | Test performance of different models on seven classification benchmarks

Dataset	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV
Molecules	**2,039**	**7,831**	**1,478**	**41,127**	**1,513**	**1,427**	**93,087**
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
GCN[17]	71.8 ± 0.9	70.9 ± 2.6	62.5 ± 2.8	74.0 ± 3.0	71.6 ± 2.0	53.6 ± 3.2	71.6 ± 4.0
GIN[18]	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
SchNet[19]	84.8 ± 2.2	77.2 ± 2.3	71.5 ± 3.7	70.2 ± 3.4	76.6 ± 1.1	53.9 ± 3.7	71.3 ± 3.0
MGCN[53]	**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[20]	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.[45]	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[44]	**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
$MolCLR_{GCN}$	73.8 ± 0.2	74.7 ± 0.8	86.7 ± 1.0	77.8 ± 0.5	78.8 ± 0.5	66.9 ± 1.2	84.0 ± 1.8
$MolCLR_{GIN}$	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**

The first seven models are supervised learning methods and the last four are self-supervised/pre-training methods. Mean and standard deviation of test ROC-AUC (%) on each benchmark are reported. Best performing supervised and self-supervised/pre-training methods for each benchmark are marked as bold.

the representations h_i and h_j into latent vectors z_i and z_j, respectively. NT-Xent loss[48] is applied to the $2n$ latent vectors z's to maximize the agreement of positive pairs while minimizing the agreement of negative ones. The framework is pre-trained on the ~10 million unlabelled data from PubChem[40].

The MolCLR pre-trained GNN models are fine-tuned for molecular property prediction as shown in Fig. 1c. Similarly to the pre-training model, the prediction model consists of a GNN backbone and an MLP head, in which the former shares the same model as the pre-trained feature extractor, and the latter maps features into the predicted molecular property. The GNN backbone in the fine-tuning model is initialized by parameter sharing from the pre-trained model while the MLP head is initialized randomly. The whole fine-tuning model is then trained in a supervised learning manner on the target molecular property database. More details can be found in the Methods section.

Molecule graph augmentation. We employ three molecule graph data augmentation strategies (Fig. 1b) for input molecules in MolCLR: 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 x_v is replaced by a mask token, m, which is distinguished from any atom features in the molecule graph shown by the red box in Fig. 1b. Through masking, the model is forced to learn the intrinsic chemical information (such as possible types of atoms connected by certain covalent bonds) within molecules.

Bond deletion. Bond deletion randomly deletes chemical bonds between the atoms with a certain ratio as the yellow box in Fig. 1b illustrates. 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. Forming and breaking of chemical bonds between atoms determines the attributes of molecules in chemical reactions[51]. Bond deletion mimics the breaking of chemical bonds, which prompts the model to learn correlations between the involvements of one molecule in various reactions.

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 neighbours of the original atom, and then the neighbours of the neighbours, until the number of masked atoms reaches a given ratio of the total number of atoms. The bonds between the masked atoms are then deleted, such that the masked atoms and deleted bonds form an induced subgraph of the original molecule graph. As the blue box in Fig. 1b shows, the removed subgraph includes all the bonds between the masked atoms. By matching the molecule graphs with different substructures removed, the model learns to find the remarkable motifs within the remaining subgraphs[52], which greatly determines the molecular properties.

Molecular property predictions. To demonstrate the effectiveness of MolCLR, we benchmark the performance on multiple challenging classification and regression tasks from MoleculeNet [24]. Details of molecular datasets can be found in Supplementary Tables 1 and 2. Table 1 shows the test area under the curve (AUC) of the receiver operating characteristic curve (ROC) (that is, ROC-AUC (%)) of our MolCLR model on classification tasks in comparison to supervised and self-supervised/pre-training baseline models. The average and standard deviation of three individual runs are reported. $MolCLR_{GCN}$ and $MolCLR_{GIN}$ denote MolCLR pre-training with GCN and GIN as feature extractors, respectively. Observations from Table 1 are as follows. (1) In comparison with other self-supervised learning or pre-training strategies, our MolCLR framework achieves the best performance on five out of seven benchmarks, with an average improvement of 4.0%. 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 best-performing supervised learning baselines, MolCLR also shows rival performance. In some benchmarks (for example, ClinTox, BACE, MUV), our pre-training model even surpasses the SOTA supervised learning methods, which include sophisticated aggregation operations or domain-specific featurization. For instance, on ClinTox, MolCLR improves the ROC-AUC by 2.7% with respect to supervised D-MPNN. (3) Notably, MolCLR performs remarkably well on datasets with a limited number of molecules, like ClinTox, BACE and SIDER. The performance validates that MolCLR learns informative representations that can be transferred among different datasets.

Table 2 includes the test performance of MolCLR and baseline models on regression benchmarks. FreeSolv, ESOL and Lipo

Table 2 | Test performance of different models on six regression benchmarks

Dataset	FreeSolv	ESOL	Lipo	QM7	QM8	QM9
Molecules	**642**	**1,128**	**4,200**	**6,830**	**21,786**	**130,829**
Tasks	**1**	**1**	**1**	**1**	**12**	**8**
SVM	3.14 ± 0.00	1.50 ± 0.00	0.82 ± 0.00	156.9 ± 0.0	0.0543 ± 0.0010	24.613 ± 0.144
GCN[17]	2.87 ± 0.14	1.43 ± 0.05	0.85 ± 0.08	122.9 ± 2.2	0.0366 ± 0.0011	5.796 ± 1.969
GIN[18]	2.76 ± 0.18	1.45 ± 0.02	0.85 ± 0.07	124.8 ± 0.7	0.0371 ± 0.0009	4.741 ± 0.912
SchNet[19]	3.22 ± 0.76	1.05 ± 0.06	0.91 ± 0.10	**74.2 ± 6.0**	0.0204 ± 0.0021	0.081 ± 0.001
MGCN[53]	3.35 ± 0.01	1.27 ± 0.15	1.11 ± 0.04	77.6 ± 4.7	0.0223 ± 0.0021	**0.050 ± 0.002**
D-MPNN[20]	2.18 ± 0.91	**0.98 ± 0.26**	**0.65 ± 0.05**	105.8 ± 13.2	**0.0143 ± 0.0022**	3.241 ± 0.119
Hu et al.[45]	2.83 ± 0.12	1.22 ± 0.02	0.74 ± 0.00	110.2 ± 6.4	0.0191 ± 0.0003	4.349 ± 0.061
N-Gram[44]	2.51 ± 0.19	**1.10 ± 0.03**	0.88 ± 0.12	125.6 ± 1.5	0.0320 ± 0.0032	7.636 ± 0.027
$MolCLR_{GCN}$	2.39 ± 0.14	1.16 ± 0.00	0.78 ± 0.01	**83.1 ± 4.0**	0.0181 ± 0.0002	3.552 ± 0.041
$MolCLR_{GIN}$	**2.20 ± 0.20**	**1.11 ± 0.01**	**0.65 ± 0.08**	87.2 ± 2.0	**0.0174 ± 0.0013**	**2.357 ± 0.118**

The first seven models are supervised learning methods and the last four are self-supervised/pre-training methods. Mean and standard deviation of test RMSE (for FreeSolv, ESOL, Lipo) or MAE (for QM7, QM8 and QM9) are reported. Best performing supervised and self-supervised/pre-training methods for each benchmark are marked as bold.

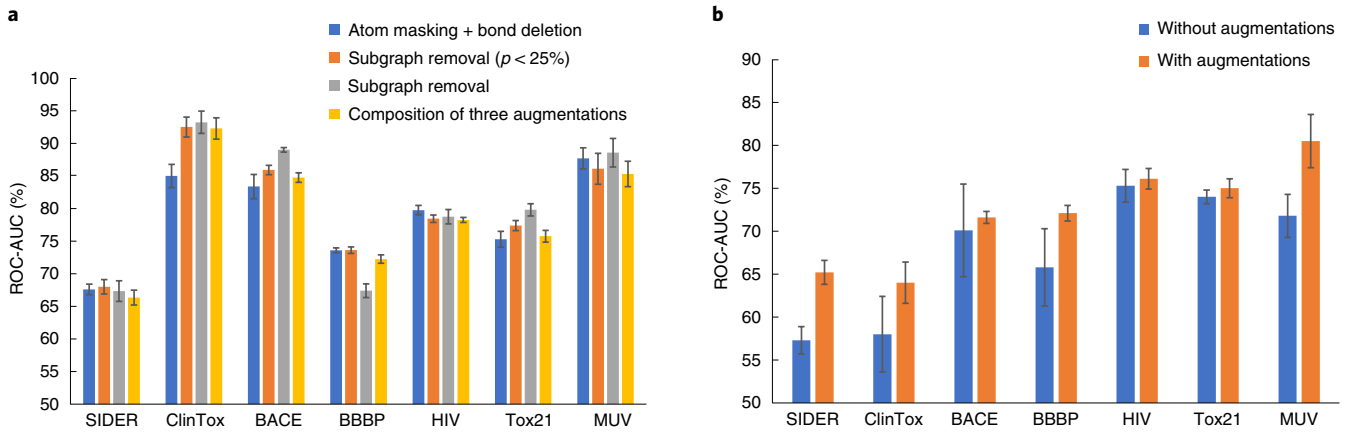

Fig. 2 | Investigation of molecule graph augmentations on classification benchmarks. a, Test performance of MolCLR models with different compositions of molecule graph augmentation strategies. **b**, Test performance of GIN models trained via supervised learning with and without molecular graph augmentations. The height of each bar represents the mean ROC-AUC (%) on the benchmark, and the length of each error bar represents the standard deviation.

use root-mean-square error (RMSE) as the evaluation metric while QM7, QM8 and QM9 are measured via mean-absolute error (MAE), following the recommendation from MoleculeNet[24]. Regression tasks are more challenging in comparison with classification since the latter only considers manually-defined discrete labels. Observations from Table 2 are the followings. (1) MolCLR surpasses other pre-training baselines in five out of six benchmarks and achieves almost the same performance on the remaining ESOL benchmark. Compared with ref. [45], which also implements GIN as the encoder, $MolCLR_{GIN}$ outperforms it on all the six regression databases. On QM7 and QM9, for example, the improvement ratios over Hu et al. are 20.9% and 45.8%, respectively. (2) In comparison with supervised learning models, MolCLR reaches competitive performance in most cases. For example, MolCLR obtains similar results as the best performing supervised D-MPNN on Lipo database. Also, GCN and GIN achieve better prediction performance via MolCLR pre-training on all regression benchmarks. Although, in QM9, MolCLR does not rival with supervised SchNet[19] and MGCN[53]. As the two models are specifically designed for quantum interaction and make use of extra 3D positional information. Notably, though SchNet and MGCN demonstrate superior performance on datasets concerning quantum mechanics properties (that is, QM7, QM8, and QM9), they do not show advantages over other supervised learning baselines on remaining benchmarks. Moreover, MolCLR pre-training is still demonstrated to be effective on the challenging QM9 benchmark. In comparison to GCN and GIN without pre-training, MolCLR still greatly boosts the performance by 38.7% and 50.3%, respectively. Also, MolCLR performs better than other self-supervised learning baselines on QM9, which validates the efficacy of MolCLR. Since properties in QM9 are of various units and magnitudes, detailed results of QM9 are reported in Supplementary Table 3.

Both Tables 1 and 2 show that MolCLR pre-training greatly improves the performance on all the benchmarks compared to supervised GCN and GIN, which demonstrates the effectiveness of MolCLR. On classification benchmarks, the average gains via MolCLR are 12.4% for GCN and 16.8% for GIN. Similarly, on regressions, the averaged improvement ratios are 27.6% for GCN and 33.5% for GCN. In general, GIN demonstrates more improvement than GCN through MolCLR pre-training. This could be because GIN has more parameters and are capable of learning more representative molecular features. Also, MolCLR shows better prediction

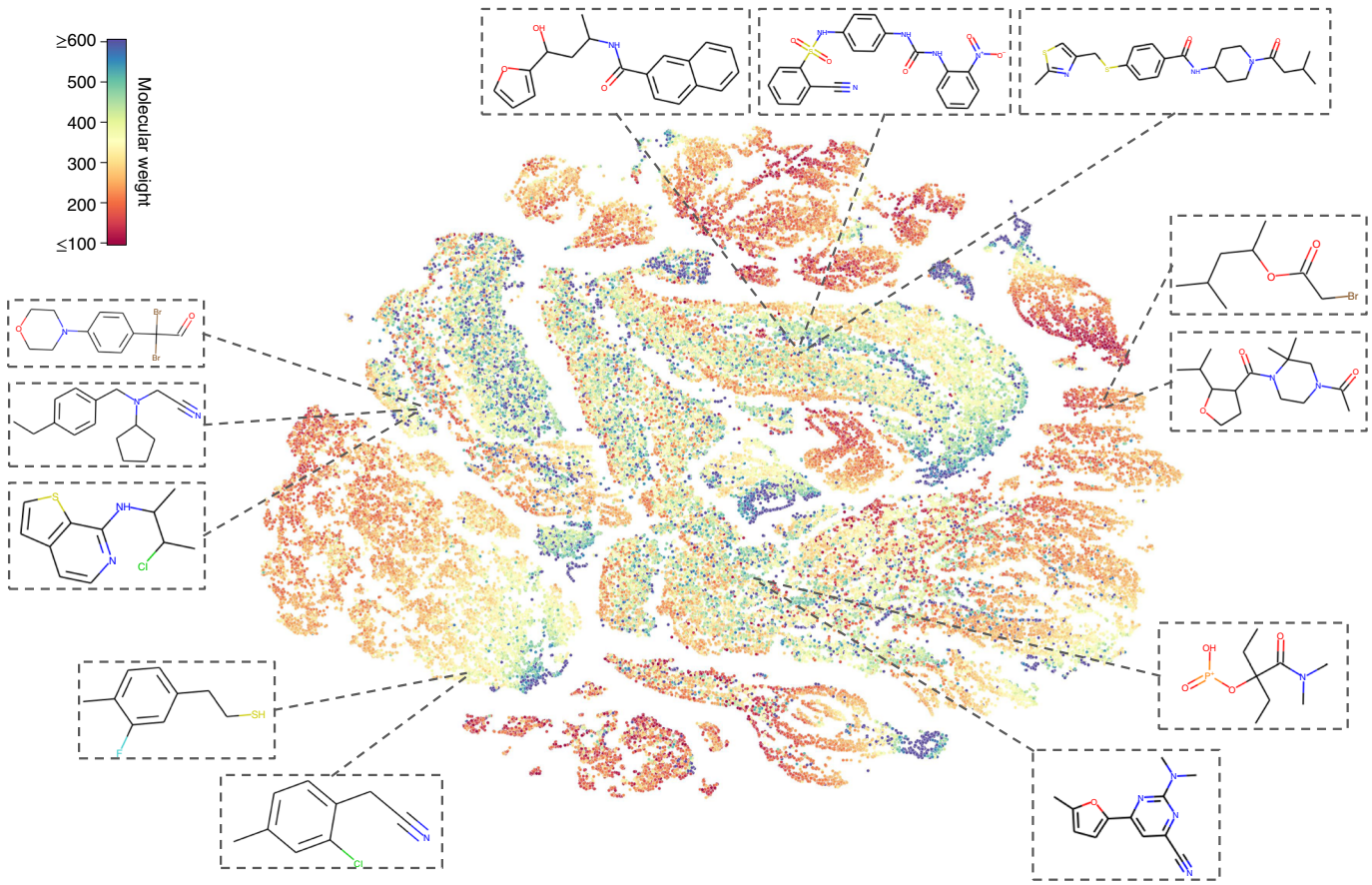

Fig. 3 | Visualization of molecular representations learned by MolCLR via t-SNE. Representations are extracted from the validation set of the pre-training dataset, which contains 100,000 unique molecules. Each point is coloured by its corresponding molecular weight (g mol^{-1}). Some molecules close in the representation domain are also shown.

accuracy in comparison to other pre-training/self-supervised learning baselines in most cases. It should be emphasized that MolCLR benefits from pre-training on large unlabelled databases while the other supervised/self-supervised learning baselines do not. Leverage of unlabelled data provides a great advantage for MolCLR over other baselines in generalizing among the chemical space and various molecular properties. Influence of the pre-training database on MolCLR is further investigated in Supplementary Table 4 and Supplementary Fig. 1. Such capability of generalization bears promise for predicting potential molecular properties in drug discovery and design.

Optimal molecule graph augmentations. To systematically analyse the effect of molecule graph augmentation strategies, we compare different compositions of atom masking, bond deletion and subgraph removal. Shown in Fig. 2a are the ROC-AUC (%) mean and standard deviation of each data augmentation strategy on different benchmarks. Four augmentation strategies 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 25%; (3) subgraph removal with a fixed 25% ratio; and (4) composition of all the three augmentation methods. Specifically, a subgraph removal with a random ratio 0% to 25% is applied at first. Then if the ratio of masked atoms is smaller than 25%, we continue to randomly mask atoms 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.

As Fig. 2a illustrates, subgraph removal with a 25% ratio reaches the best performance on average among all the four compositions. Its outstanding performance can be attributed to the fact that subgraph removal is an intrinsic combination 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 the BBBP dataset because the 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 (strategy 4) hurts the ROC-AUC compared with single subgraph removal augmentation in most benchmarks. A possible reason is that 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. In general, subgraph removal achieves superior performance in most benchmarks. However, it is also indicated that the optimal molecule graph augmentation is task-independent.

Molecule graph augmentation on supervised learning. The molecule graph augmentation strategies in our work, namely atom masking, bond deletion and subgraph removal, can be implemented as a generic 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 and without augmentations from random initialization. Specifically, subgraph masking with a fixed ratio, 25%, is implemented as the augmentation. Figure 2b 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

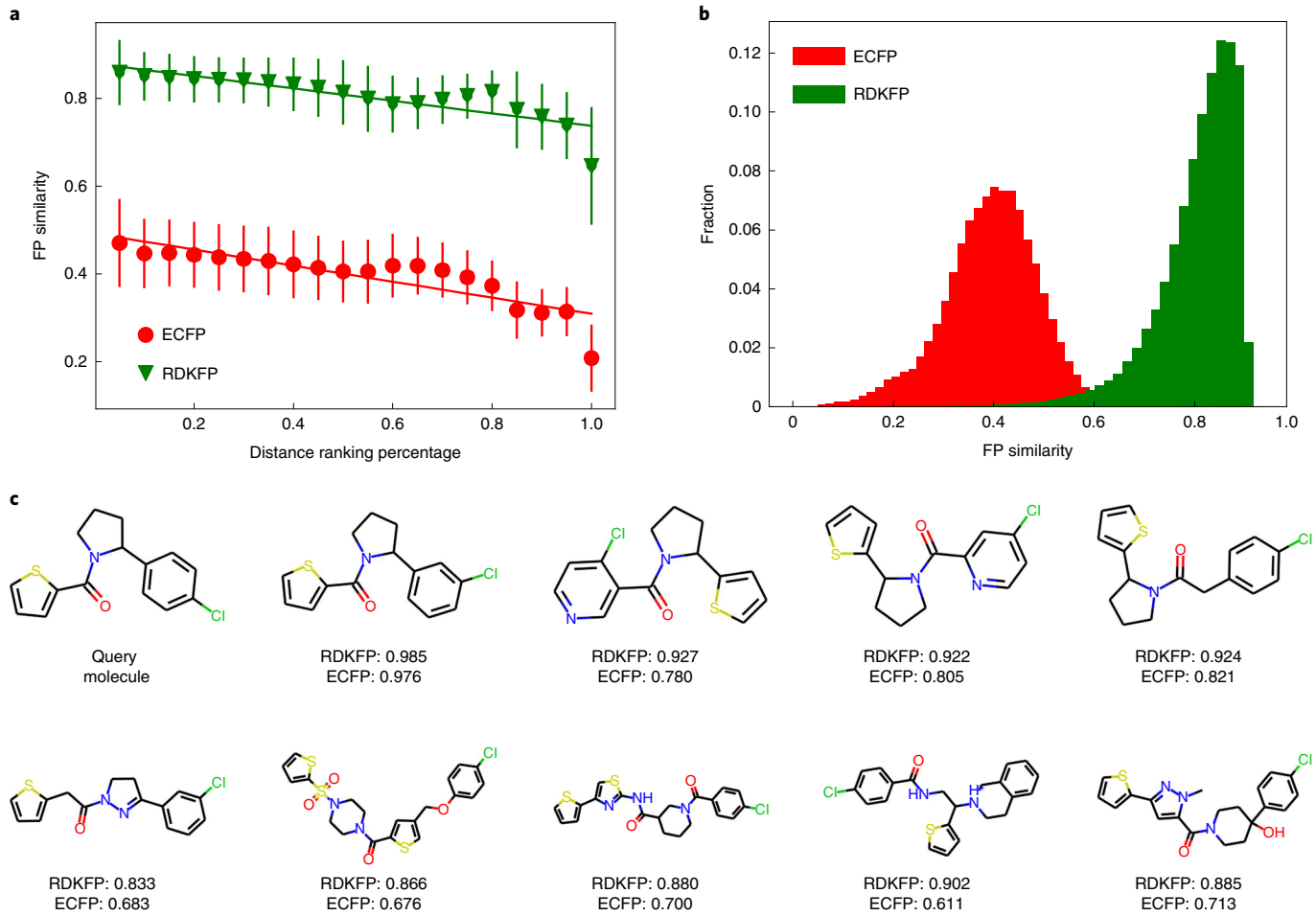

Fig. 4 | Comparison of MolCLR-learned representations and conventional FPs using the query molecule (PubChem ID 42953211). a, Change of ECFP and RDKFP similarities with respect to the distance between MolCLR representations. **b**, Distribution of ECFP and RDKFP similarities with the query molecule. **c**, The query molecule and nine closest molecules in MolCLR representation domain with RDKFP and ECFP similarities labelled.

augmentations. Molecule graph augmentations improve the averaged ROC-AUC score by 7.2%. Implementation of our molecule graph augmentation strategies on supervised molecular property prediction tasks improves the performance greatly even without pre-training. It is indicated that molecule graph augmentations are effective in helping GNNs learn robust and representative features. For instance, subgraph removal matches partially observed molecule graphs. Therefore, the model learns to find the remarkable motifs within the remaining subgraphs, which greatly benefits molecular property learning.

Investigation of MolCLR representation. We examine the representations learned by pre-trained MolCLR using t-SNE embedding[54]. The t-SNE algorithm maps close molecular representations to adjacent points in 2D. Figure 3 shows 100,000 molecules from the validation set of the PubChem database embedded to 2D via t-SNE, coloured 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 Fig. 3, 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 illustrates that even without labels, the model learns intrinsic connections between molecules as molecules with similar properties have close features. More visualizations of MolCLR representations can be found in Supplementary Fig. 2.

To further evaluate MolCLR, we compare the MolCLR-learned representations with conventional molecular FPs, for example, ECFP[5] and RDKFP. In particular, given a query molecule, we extract its representation via MolCLR and calculate its cosine distances with all reference molecules in our pre-training database. Cosine distance between two representations (u, v) are defines as $1 - \frac{u \cdot v}{\|u\|\|v\|}$. All reference molecules are then ranked by the representation distances and uniformly divided into 20 bins based on the ranking percentage. The lower the percentage threshold is, the more similar molecules are expected with respect to the query, as the MolCLR representations are closer. Within each bin, 5,000 molecules are randomly selected and their dice FP similarities with the query are calculated. Figure 4 shows an example of a query molecule (PubChem ID 42953211). Shown in Fig. 4a are the mean and standard deviation of FP similarities within each bin. The distribution of similarities using both ECFP and RDKFP are shown in Fig. 4b. ECFP tends to obtain lower similarities than RDKFP since the former covers a wider range of features relevant to molecular activity. It is shown, though, as the MolCLR representation distance increases, both the ECFP and RDKFP similarities decrease. The averaged RDKFP similarities at the top 5% is ~0.9 and drops to ~0.67 at the last 5%. Similarly, the averaged ECFP similarity drops from ~0.49 at the top 5% to ~0.21 at the last 5%. Though there are fluctuations as

the percentage threshold increases, the overall tendencies are consistent among the MolCLR learned representations and chemical FPs. Namely, the distance between MolCLR representations effectively reflects the molecular similarity. Besides, nine molecules that are closest to the query molecule in the MolCLR representation domain are illustrated in Fig. 4c with both FPs similarities labelled. These molecules share high RDKFP similarities from 0.833 to 0.985, which further demonstrate MolCLR learns chemically meaningful representations. It is observed that these selected molecules share the same functional groups, including alkyl halides (chlorine), tertiary amines, ketones and aromatics. A thiophene structure can also be found in all the molecules. Notably, the second molecule in the first row in Fig. 4c is exactly the same as the query molecule except for the position of the chlorine, hence the highest similarities. It is indicated that through contrastive learning on large unlabelled data, MolCLR automatically embeds molecules to representative features and distinguishes the compounds in a chemically reasonable manner. More examples of query molecules can be found in Supplementary Fig. 3.

Conclusion

In this work, we investigate self-supervised learning for molecular representation. Specifically, we propose MolCLR via GNNs and three molecular graph augmentation 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 demonstrate the transferability to molecular tasks with limited data and the power of generalization on the large chemical space. There are many promising directions to investigate as future works. For instance, improvement of the GNN backbones (for example, transformer-based GNN architectures[55]) can help extract better molecular representations. Besides, visualization and interpretation of self-supervised learned representations are of great interest[56]. Such investigations can help researchers better understand chemical compounds and benefit drug discovery.

Methods

Graph neural networks. In our work, a molecule graph G is defined as $G=(V,E)$, where V and E are nodes (atoms) and edges (chemical bonds), respectively[57]. Modern GNNs utilize a neighbourhood aggregation operation, which updates the node representation iteratively[17]. The aggregation update rule for a node feature on the kth layer of a GNN is given in equation (1):

$$\begin{aligned} a_v^{(k)} &= \text{AGGREGATE}^{(k)}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}), \\ h_v^{(k)} &= \text{COMBINE}^{(k)}(h_v^{(k-1)}, a_v^{(k)}), \end{aligned} \tag{1}$$

where $h_v^{(k)}$ is the feature of node v at the kth layer and $h_y^{(0)}$ is initialized by node feature x_v. $\mathcal{N}(v)$ denotes the set of all the neighbours of node v. To further extract a graph-level feature h_G, readout operation integrates all the node features among the graph G as given in equation (2):

$$h_G = \text{READOUT}(\{h_u^{(k)} : v \in G\}). \tag{2}$$

In our work, we build GNN encoders based on GCN[17] and GIN[18]. GCN integrates the aggregation and combination operations by introducing a mean pooling over the node itself and its adjacencies before the linear transformation. While GIN utilizes an MLP and weighted summation of node features in the aggregation. Both are simple yet generic graph convolutional operations. Additionally, we implement widely used mean pooling as the readout.

Contrastive learning. Contrastive learning[58] aims at learning representation through contrasting positive data pairs against negative pairs. SimCLR[48] demonstrates that contrastive learning for images can greatly benefit from the composition of data augmentations and large batch sizes. Based on InfoNCE loss[47], SimCLR proposes the NT-Xent loss as given in equation (3):

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}\{k \neq i\} \exp(\text{sim}(z_i, z_k)/\tau)}, \tag{3}$$

where z_i and z_j are latent vectors extracted from a positive data pair, N is the batch size, sim$(\cdot)$ measures the similarity between the two vectors, and τ is the temperature parameter. In our MolCLR, we follow the NT-Xent loss to conduct pre-training on GNN encoders and implement cosine similarity as $\text{sim}(z_i, z_j) = \frac{z_i^T z_j}{\|z_i\|_2 \|z_j\|_2}$. Further investigation of τ on MolCLR pre-training is included in Supplementary Table 5. Though contrastive learning frameworks have been implemented to various domains, including unstructured graphs[46], sentence embeddings[59] and robotics planning[60]. Contrastive learning has not yet been investigated comprehensively and elaborately for molecule graphs.

Datasets. *Pre-training dataset.* For MolCLR pre-training, we use approximately 10 million unique unlabelled molecule SMILES collected by ChemBERTa[42] from PubChem[40]. RDKit[61] is then utilized to build the molecule graphs and extract chemical features 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 13 datasets from MoleculeNet[24], containing 44 binary classification tasks and 24 regression tasks in total. These tasks cover molecular properties of multiple domains. For all datasets except QM9, we use the scaffold split to create an 80/10/10 train/valid/test split as suggested in ref. [45]. Unlike the common random split, the scaffold split, which is based on molecular substructures, makes the prediction task more challenging yet realistic. QM9 follows the random splitting setting as implementations of most related works[19,44,53] for comparison.

Training details. Each atom on the molecule graph is embedded by its atomic number and chirality type, while each bond is embedded by its type and direction. We implement a five-layer graph convolution[17,18] with ReLU activation as the GNN backbone, and follow the modification reported by Hu et al.[45] to make aggregations 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[62] optimizer with weight decay 10^{-5} is used to optimize the NT-Xent loss. After the initial 10 epochs with a learning rate, 5×10^{-4}, a cosine learning decay is implemented. The model is trained with batch size 512 for the total 50 epochs.

For the downstream task fine-tuning, we add a randomly initialized MLP on top of the base GNN feature extractor. Softmax cross-entropy loss and ℓ_1 loss are implemented for classification and regression tasks, respectively. On each task, we conduct 100-epoch fine-tuning of the pre-trained model three times to get the average and standard deviation of performance on the test set. We train the model on the training set only and perform search of hyper-parameters on the validation set for the best results. The whole framework is implemented based on Pytorch Geometric[63]. More fine-tuning details are included in Supplementary Table 6.

Baselines. *Supervised learning models.* We comprehensively evaluate the performance of our MolCLR model in comparison with supervised learning methods. For shallow machine learning models, Random Forest[64] and Support Vector Machine[65] are implemented, which take molecular FPs as the input. Multiple GNNs are also included. GCN[17] and GIN[18,45] with edge feature involved in aggregation are considered. Besides, several GNN models that achieve SOTA on several molecular benchmarks are implemented as baselines, that is, SchNet[19], MGCN[53] and D-MPNN[20]. These GNNs are designed specifically for molecular. For example, SchNet and MGCN explicitly model quantum interactions within molecules.

Self-supervised learning models. To better demonstrate the effectiveness of MolCLR framework, we further include other pre-training or self-supervised learning models as baselines. Hu et al.[45] propose both node-level and graph-level pre-training for molecule graphs. It should be pointed out that though node-level pre-training is based on self-supervision, while the graph-level pre-training is supervised on some molecular property labels[45]. N-Gram graph[44] is also implemented, which computes a compact representation directly through the molecule graph.

Data availability

The pre-training data and molecular property prediction benchmarks used in this work are available in the both the CodeOcean capsule at https://doi.org/10.24433/CO.8582800.v1[49] and the GitHub repository at https://github.com/yuyangw/MolCLR.

Code availability

The code accompanying this work is available in both the CodeOcean capsule at https://doi.org/10.24433/CO.8582800.v1[49] and the GitHub repository at https://github.com/yuyangw/MolCLR.

Received: 14 June 2021; Accepted: 14 January 2022;
Published online: 3 March 2022

References

1. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
2. Huang, B. & Von Lilienfeld, O. A. Communication: Understanding molecular representations in machine learning: the role of uniqueness and target similarity. *J. Chem. Phys.* **145**, 161102 (2016).
3. David, L., Thakkar, A., Mercado, R. & Engkvist, O. Molecular representations in AI-driven drug discovery: a review and practical guide. *J. Cheminform.* **12**, 56 (2020).
4. Oprea, T. I. & Gottfries, J. Chemography: the art of navigating in chemical space. *J. Comb. Chem.* **3**, 157–166 (2001).
5. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
6. Duvenaud, D. et al. Convolutional networks on graphs for learning molecular fingerprints. In *Proc. 28th International Conference on Neural Information Processing Systems* 2224–2232 (MIT Press, 2015).
7. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning* 1263–1272 (PMLR, 2017).
8. Karamad, M. et al. Orbital graph convolutional neural network for material property prediction. *Phys. Rev. Mater.* **4**, 093801 (2020).
9. Chmiela, S., Sauceda, H. E., Müller, K.-R. & Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **9**, 3887 (2018).
10. Deringer, V. L. et al. Realistic atomistic structure of amorphous silicon from machine-learning-driven molecular dynamics. *J. Phys. Chem. Lett.* **9**, 2879–2885 (2018).
11. Wang, W. & Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Comput. Mater.* **5**, 125 (2019).
12. Altae-Tran, H., Ramsundar, B., Pappu, A. S. & Pande, V. Low data drug discovery with one-shot learning. *ACS Cent. Sci.* **3**, 283–293 (2017).
13. Magar, R., Yadav, P. & Farimani, A. B. Potential neutralizing antibodies discovered for novel corona virus using machine learning. *Sci. Rep.* **11**, 5261 (2021).
14. Wang, Y., Cao, Z. & Farimani, A. B. Efficient water desalination with graphene nanopores obtained using artificial intelligence. *npj 2D Mater. Appl.* **5**, 66 (2021).
15. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comp. Sci.* **28**, 31–36 (1988).
16. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (selfies): a 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
17. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations* (2017).
18. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations* (2019).
19. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
20. Yang, K. et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **59**, 3370–3388 (2019).
21. Kirkpatrick, P. & Ellis, C. Chemical space. *Nature* **432**, 823 (2004).
22. Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **16**, 3–50 (1996).
23. Brown, N., Fiscato, M., Segler, M. H. & Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2019).
24. Wu, Z. et al. Moleculenet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
25. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
26. Vamathevan, J. et al. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **18**, 463–477 (2019).
27. Unterthiner, T. et al. Deep learning as an opportunity in virtual screening. In *Proc. Deep Learning Workshop at NIPS* Vol. 27 (2014).
28. Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E. & Svetnik, V. Deep neural nets as a method for quantitative structure–activity relationships. *J. Chem. Inf. Model.* **55**, 263–274 (2015).
29. Ramsundar, B. et al. Massively multitask networks for drug discovery. Preprint at https://arxiv.org/abs/1502.02072 (2015).
30. Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. Grammar variational autoencoder. In *International Conference on Machine Learning* 1945–1954 (PMLR, 2017).
31. Gupta, A. et al. Generative recurrent networks for de novo drug design. *Mol. Inf.* **37**, 1700111 (2018).
32. Xu, Z., Wang, S., Zhu, F. & Huang, J. Seq2seq fingerprint: an unsupervised deep molecular embedding for drug discovery. In *Proc. 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* 285–294 (ACM, 2017).
33. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
34. Schwaller, P. et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS Cent. Sci.* **5**, 1572–1583 (2019).
35. Maziarka, Ł. et al. Molecule attention transformer. Preprint at https://arxiv.org/abs/2002.08264 (2020).
36. Feinberg, E. N. et al. PotentialNet for molecular property prediction. *ACS Cent. Sci.* **4**, 1520–1530 (2018).
37. Klicpera, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. Preprint at https://arxiv.org/abs/2003.03123 (2020).
38. Gaulton, A. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **40**, D1100–D1107 (2012).
39. Sterling, T. & Irwin, J. J. Zinc 15–ligand discovery for everyone. *J. Chem. Inf. Model.* **55**, 2324–2337 (2015).
40. Kim, S. et al. Pubchem 2019 update: improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109 (2019).
41. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. Preprint at https://arxiv.org/abs/1810.04805 (2018).
42. Chithrananda, S., Grand, G. & Ramsundar, B. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. Preprint at https://arxiv.org/abs/2010.09885 (2020).
43. Wang, S., Guo, Y., Wang, Y., Sun, H. & Huang, J. SMILES-BERT: large scale unsupervised pre-training for molecular property prediction. In *Proc. 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* 429–436 (ACM, 2019).
44. Liu, S., Demirel, M. F. & Liang, Y. N-gram graph: simple unsupervised representation for graphs, with applications to molecules. In *Thirty-third Conference on Neural Information Processing Systems* (NeurIPS, 2019).
45. Hu, W. et al. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations* (2020).
46. You, Y. et al. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **33**, 5812–5823 (2020).
47. van den Oord, A., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. Preprint at https://arxiv.org/abs/1807.03748 (2018).
48. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning* 1597–1607 (PMLR, 2020).
49. Wang, Y., Wang, J., Cao, Z. & Farimani, A. B. MolCLR: molecular contrastive learning of representations via graph neural networks. *CodeOcean* https://doi.org/10.24433/CO.8582800.v1 (2021).
50. Chen, T., Kornblith, S., Swersky, K., Norouzi, M. & Hinton, G. Big self-supervised models are strong semi-supervised learners. Preprint at https://arxiv.org/abs/2006.10029 (2020).
51. Do, K., Tran, T. & Venkatesh, S. Graph transformation policy network for chemical reaction prediction. In *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 750–760 (ACM, 2019).
52. Jin, W., Barzilay, R. & Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning* 4839–4848 (PMLR, 2020).
53. Lu, C. et al. Molecular property prediction: a multilevel quantum interactions modeling perspective. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 33, 1052–1060 (AAAI, 2019).
54. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
55. Yun, S., Jeong, M., Kim, R., Kang, J. & Kim, H. J. Graph transformer networks. In *Advances in Neural Information Processing Systems* Vol. 32 (eds. Wallach, H. et al.) (Curran Associates, 2019).
56. Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E. & Hoffmann, H. Explainability methods for graph convolutional neural networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10772–10781 (IEEE, 2019).
57. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. & Vandergheynst, P. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process. Mag.* **34**, 18–42 (2017).
58. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* 9729–9738 (IEEE, 2020).
59. Gao, T., Yao, X. & Chen, D. SimCSE: simple contrastive learning of sentence embeddings. Preprint at https://arxiv.org/abs/2104.08821 (2021).
60. Wang, J., Lu, Y. & Zhao, H. CLOUD: contrastive learning of unsupervised dynamics. Preprint at https://arxiv.org/abs/2010.12488 (2020).
61. Landrum, G. RDKit: open-source cheminformatics (2006); https://www.rdkit.org/
62. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).

63. Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).
64. Ho, T. K. Random decision forests. In *Proc. 3rd International Conference on Document Analysis and Recognition* Vol. 1, 278–282 (IEEE, 1995).
65. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).

Acknowledgements

We thank the start-up fund provided by the Department of Mechanical Engineering at Carnegie Mellon University. The work is also funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), US Department of Energy, under award no. DE-AR0001221.

Author contributions

Y.W., J.W. and A.B.F. designed the research study. Y.W., J.W. and Z.C. developed the method, wrote the code and performed the analysis. All authors wrote and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00447-x.

Correspondence and requests for materials should be addressed to Amir Barati Farimani.

Peer review information *Nature Machine Intelligence* thanks Alán Aspuru-Guzik and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.