POINT-BASED MOLECULAR REPRESENTATION LEARNING FROM CONFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular representation learning (MRL) aims to embed molecules into vectors in a high dimensional latent space, which can be used (and reused) for the prediction of various molecular properties. Most current MRL models exploited the SMILES (Simplified Molecular-Input Line-Entry System) strings or molecular graphs as the input format of molecules. As a result, these methods may not capture the full information encoded in the three-dimensional (3D) molecular conformations (also known as the *conformers*). With mature algorithms for generating 3D molecular conformers, we propose to engage the abundant geometric information in the molecular conformers by representing molecules as *point sets*, and adapt the point-based deep neural network for MRL. Specifically, we designed an atom-shared elemental operation that extracts features from individual atoms as well as atomic interactions (including covalent bonds and non-covalent interactions), and a mini-network that ensures the representation invariant to rotations and translations of the molecular conformers. We trained the deep neural network (referred to as *Mol3DNet*) for a variety of tasks of molecular properties prediction using benchmarking datasets. The experimental results demonstrated that Mol3DNet achieves state-of-the-art performance on these classification and regression tasks, except for one task (solubility prediction) where all deep learning models underperform a customized machine learning model.

1 INTRODUCTION

How to represent molecules is a fundamental problem in computational molecular science. Traditionally, molecules are represented in intuitive formats such as chemical formulas, structural formulas, or skeletal formulas. However, these formats cannot be directly input into computers. To encode molecules in computers, researchers designed the string representations, e.g., the commonly used SMILES (Simplified Molecular-Input Line-Entry System) (Weininger, 1988) strings, and the graph representations, e.g., the molecular graph (McNaught et al., 1997). Both methods were used in molecular representation learning (MRL), which aims to learn a high dimensional vector representation of molecules (Li et al., 2001) using deep neural networks (DNNs). The learned vectors can be used for various prediction tasks in molecular science, such as the prediction of chemical properties (Wu et al., 2018; Zhu et al., 2022), chemical reactions (Fooshee et al., 2018), the interface (Fout et al., 2017) and affinity (Wang et al., 2021b) of protein–ligand interactions, and putative drugs (Lavecchia, 2019; Gentile et al., 2020), etc.

SMILES-based methods. The SMILES-based MRL methods consider each SMILES string as a sequence, and then exploited natural language processing (NLP) models (e.g. Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2018)) to learn latent representations from sequences. For instance, Molecular Transformer (Schwaller et al., 2019; Pesciullesi et al., 2020), MolBERT (Fabian et al., 2020) and SMILES-BERT (Wang et al., 2019) were developed for MRL and the prediction of molecular properties. Despite some successes, because SMILES encodes only partial structural information of molecules, these methods often encounter the bottleneck in improving the performance of prediction tasks (Li et al., 2018).

Graph-based methods. Many of the recent MRL methods exploit the molecular graph, a graph representation of molecules in which the atoms in a molecule are represented as nodes and the covalent bonds are presented as edges. Molecular graphs can be processed by different kinds of graph neural

networks (GNNs), such as GCN (Kipf & Welling, 2016) or GIN (Xu et al., 2018). Built upon these model architectures, different operations were designed to automatically extract molecular features (e.g., neural graph fingerprints (NFP) (Duvenaud et al., 2015)) that simulate the topological fingerprints such as the extended-connectivity circular fingerprints (ECFP) (Rogers & Hahn, 2010) for molecular characterization. Several recent researches attempted to incorporate additional structural information for MRL. For instance, DimeNet (Gasteiger et al., 2019) and DimeNet++ (Klicpera et al., 2020) performed message passing utilizing the bond angles and lengths, which is embedded by spherical Bessel functions and spherical harmonics. SphereNet (Liu et al., 2021b) proposed spherical message passing to learn features from molecular graphs with bond lengths, angles and torsions. GraphMVP (Jing et al., 2021) applied self-supervised learning to train 2D GNN and 3D GNN together, aiming to improve 2D MRL by complementary 3D geometric information. MolR (Wang et al., 2021a) integrated chemical reaction information into the learning of the molecular graph embedding. Compared with the SMILES-based methods, the graph-based methods exploited 2D topological information as well as some 3D geometric information for MRL. Nevertheless, in these models, only the local geometric information related to the covalent interactions (i.e., bond lengths, angles and torsions) were considered and encoded as the attributes of the respective atoms, and thus the 3D structural information may not be fully captured during the learning process.

Point-based methods. Finally, following the convention in quantum mechanics and molecular dynamics, a chemical molecule can also be encoded as a set of atomic points, each with a 3D coordinate. The covalent bonds between atoms do not need to be encoded explicitly because they are attributed by the overlap between the atomic orbitals, and can be inferred from the types and 3D coordinates of respective atoms. In principle, the point-based representation captures the complete structural information about the molecule, and thus serves as the adequate input for MRL. In practice, however, very few point-based DNN models were developed specifically for MRL. SchNet (Schütt et al., 2017) is the most known model in this category, which used continuous-filter convolutional layers to capture the subtle positional changes among the atoms in a molecule by extracting features from the pairwise distances between atoms. However, the SchNet model does not explicitly utilize the coordinates of atoms, and neglected the direction of atomic interactions, which contains important information about the bond angles and the topology of molecular substructures.

In this paper, we propose to encode a molecule as a *points set* (or a *points cloud*), which allows for the direct representation of the molecule's 3D conformation as well as the learning of interactions between each atom and its nearest neighbors. We designed the architecture of the deep neural network by integrating essential components in two baseline models, PointNet (Qi et al., 2017) and DGCNN (Manessi et al., 2020), which were designed to address 3D computer vision problems, but have not been applied to MRL and molecular properties prediction. To adapt these models for our purpose, we revised the model in two aspects: 1) the points in a molecular point cloud are annotated by not only their x, y, z-coordinates, but also atomic attributes (such as the type and mass of the atoms); and 2) the molecular point cloud accepts only the rotation and translation as invariant transformations (i.e., under these transformations, the molecular point clouds are considered as the equivalent inputs), but not other *affine* transformations such as dilations and shears that are accepted in a conventional point cloud model for computer vision tasks. To implement these adaptions, we designed an elemental operation, **MolAConv**, to extract features from the attributes of individual atoms as well as their neighbors, and modified the transformation network into **TnRNet**, in which only rotation and translation transformations are accepted. We show that point-based methods achieve comparable performance with graph-based methods, while our point-based model for MRL named **Mol3DNet** performs state-of-the-art in both classification and regression tasks except for the solubility prediction where all deep learning models underperform a customized machine learning model.

2 Background

2.1 Generating 3D molecular conformations

The 3D conformation of a chemical compound can be determined by experimental methods, e.g., X-ray Crystallography or Nuclear Magnetic Resonance (NMR). However, these modern technologies are expensive and time-consuming. Hence, computational methods were developed to simulate the 3D conformations (i.e., the *conformers*) automatically from the 2D molecular graphs. The

Model	Input of Each Layer	Elemental Operation (h)
PointNet	$x_i^{(l)}$ (coordinates of point-i in layer-l)	$x_i^{(l+1)} = \text{MLP}(x_i^{(l)})$
DGCNN	$x_i^{(l)}$ (coordinates of point-i in layer-l)	$x_i^{(l+1)} = \max_{j \in \mathcal{N}(x_i^{(l)})} \text{MLP}(x_j^{(l)} - x_i^{(l)}, x_i^{(l)})$ where $\mathcal{N}(x_i^{(l)})$ denotes the k-nearest neighbors of $x_i^{(l)}$
SchNet	r_i (coordinates of point-i) $z_i^{(l)}$ (features of point-i in layer-l)	$z_i^{(l+1)} = \text{MLP}(\sum_{j=1}^{n} \text{CF}(x_j^{(l)}, r_i, r_j))$ $\text{CF}(x_j, r_i, r_j) = x_j \cdot \exp(-\gamma \lVert \lVert r_i - r_j \rVert - \mu \rVert^2)$ where γ and μ are parameters of continuous-filter

Table 1: The elemental operations of point-based deep neural networks. The input $x_i^{(l)}$ and $z_i^{(l)}$ are dynamic among layers, while r_i is constant.

Experimental-Torsion with basic Knowledge Distance Geometry (ETKDG) (Riniker & Landrum, 2015) is a commonly used rule-based conformer generation algorithm, which is implemented in the open-source `RDKit` package. In the latest version (ETKDGv3) (Wang et al., 2020), the algorithm was further improved for the conformer generation of molecules containing small or large aliphatic (i.e., non-aromatic) rings. In addition to ETKDG, several commercial software tools based on more accurate knowledge-based algorithms are available. Implemented in the `OpenEye` package, OMEGA (Hawkins et al., 2010; Hawkins & Nicholls, 2012) used GPU to accelerate computation. More recently, machine learning models were also developed for 3D conformer generation. For instance, GraphDG (Simm & Hernández-Lobato, 2019) exploited a probabilistic model, while GeoMol (Ganea et al., 2021) exploited a message passing neural networks (MPNNs) to generate statistically independent samples of molecular graphs. Notably, machine learning models often rely heavily on the training data and the tuning of hyper-parameters, and are not obviously better than conventional computational methods. Therefore, in this study, we used the conventional computational methods, specifically, the open source RDKit package and the commercial method OMEGA, to generate 3D conformers used as the input to our model Mol3DNet.

2.2 Point-based Deep Neural Networks

The point-based deep neural networks (DNNs) should follow two invariance principles: *permutation invariance* and *transformation invariance*. PointNet (Qi et al., 2017) is the first to introduce DNNs on point clouds. It applies a multi-layer perceptron (MLP) to each point, named shared-MLP, so that the model prediction is independent on the order of the points in the input point cloud, which is distinct from the models for learning on the structured inputs like images and sequences. In addition, a mini-network is designed to predict the *affine transformation matrix*, which are then incorporated into the model to ensure the affine transformations (e.g., rotation, translation, dilation, and shear) of the input point cloud do not affect the prediction outcome of the model. PointNet does not explicitly model the interactions among the points in the point cloud. To address this issue, DGCNN (Manessi et al., 2020) used an MLP to extract the pairwise features between each point and its neighbors. Our model attempts to combine these two approaches to extract the features from each individual atom as well as its geometrically close neighbors (based on their atomic attributes).

Consider a point set with n points, denoted as $X = \{x_1, x_2, ..., x_n\} \subseteq \mathbb{R}^F$. In the point-based DNNs, each layer operates on the output point set from the previous layer, and thus, F has the same dimensions as the output from the previous layer, which varies for different layers. For point clouds used in computer vision, each element (point) is typically represented by the x, y, z-coordinates in the 3D Euclidean space ($F = 3$) at the first layer, while in the molecular point sets each element (atom) is represented by atomic attributes in addition to the atom's x, y, z-coordinates in the 3D conformers ($F = 21$ as shown in Table 6).

Following PointNet, the general idea of permutation invariance in feature extraction is to apply a symmetric function to the elements in the point set:

$$f(\{x_1, x_2, ..., x_n\}) \approx g(h(x_1), h(x_2), ..., h(x_n)) \tag{1}$$

where $f : 2^{\mathbb{R}^F} \to \mathbb{R}$, $h : \mathbb{R}^F \to \mathbb{R}^K$ and $g : \underbrace{\mathbb{R}^K \times ... \times \mathbb{R}^K}_{n} \to \mathbb{R}$.

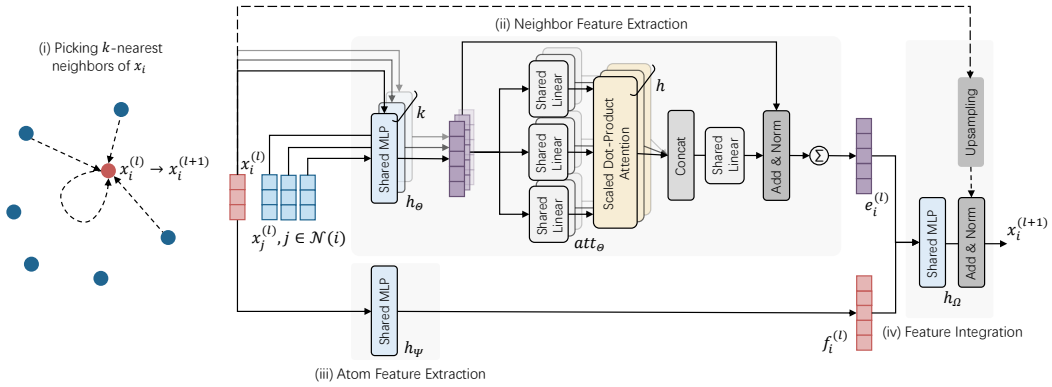

Figure 1: The Elemental operation MolAConv. MolAConv consists of three subnetworks for the feature extraction and integration in four steps: (i) for atom i with the input attribute vector x_i, a local subgraph is built containing its k-nearest neighbors, whose attribute vectors are denoted by $x_j, j \in \mathcal{N}(x_i)$; (ii) through the neighbor feature extraction subnetwork (h_Θ and att_Θ), the neighbor features are derived from x_i and x_j, and then concatenated to derive the neighbor feature vector e_i by applying the aggregate operation ($\sum$); (iii) through the atom feature extraction subnetwork (h_Ψ), the atom feature vector is derived from the atom attributes x_i; and (iv) finally, through the feature integration subnetwork (h_Ω), the feature vectors of the atom and its neighbors are integrated into a latent feature vector $x_i^{(l+1)}$ with a shortcut connection.

Usually, g can be concretized by using max-pooling ($\max$), summarizing ($\sum$) or other symmetric operations, after which the results will be the same no matter which order of the elements are in the input. The elemental operations (h) of PointNet, DGCNN and SchNet are summarized in Table 1.

3 Methodology

3.1 Elemental Operation

In the existing deep learning methods on point clouds, only the x, y, z-coordinates are taken into consideration. Unlike point clouds in computer vision, atomic attributes are crucial for MRL and molecular properties prediction. Hence, we designed the elemental operation **MolAConv** (Molecular Attentional Convolution as shown in Figure 1), aiming to extract both the geometric and atomic attributes which are summarized in Table 6. The model can be easily extended if additional attributes need to be incorporated.

For neighbor features, we modified the elemental operation of DGCNN by leveraging multi-head attention assisting the model in paying various concentrations to neighbors. In DGCNN, the features between atom pairs (i.e., an atom and one of its neighbors) are extracted by the displacement vector, $x_j - x_i$. In MRL, however, we need to incorporate the non-coordinate attributes of the atoms: for example, the displacement on the dimensions of the one-hot encoding of the atom types (Table 6) is meaningless. Hence, in MolAConv, we concatenate x_j and x_i as the input to the neighbor feature extraction subnetwork in order for the network to learn the integrative geometric and atomic features. Here, the multi-head self-attention is modified such that it could be shared by the atoms, which guarantees the *permutation invariance*. Besides, in our model, we used the $\sum$ (instead of max-pooling in DGCNN) as the aggregate function to make full use of all neighbors' features, Therefore, the neighbor features $e_i^{(l)}$ is computed by:

$$e_i^{(l)} = \sum_{j \in \mathcal{N}(x_i^{(l)})} att_\Theta(h_\Theta(x_i^{(l)}, x_j^{(l)})) \tag{2}$$

where att_Θ is the atom-shared multi-head self-attention:

$$\text{Scaled Dot-Product Attention}(V, K, Q) = V \cdot \text{Softmax}(K^\mathsf{T} \cdot Q) \tag{3}$$

$$\begin{cases} \text{Head}_1 = & \text{Scaled Dot-Product Attention}(W_1^v x, W_1^v x, W_1^v x) \\ \text{Head}_2 = & \text{Scaled Dot-Product Attention}(W_2^v x, W_2^v x, W_2^v x) \\ & \dots \\ \text{Head}_h = & \text{Scaled Dot-Product Attention}(W_h^v x, W_h^v x, W_h^v x) \end{cases} \tag{4}$$

$$att_\Theta(x) = W^o \cdot [\text{Head}_1, \text{Head}_2, ..., \text{Head}_h] \tag{5}$$

Here, the input is one atom feature $x \in \mathbb{R}^{d\times k}$, and W_i^v, W_i^k, W_i^q $(i = 1, 2, ..., h)$ and W^o are the shared weights among all atoms (i.e., the *shared linear*), and thus $W_i^v x$, $W_i^k x$, and $W_i^q x$ are not affected by the permutation of atoms. The shared linear can be easily implemented by the convolution with a 1×1 kernel similar to the shared MLP (Lin et al., 2013). When implementing the scaled dot-product attention shared by all atoms, the input is the features of all atoms, $X \in \mathbb{R}^{d\times n\times k}$, where d is the number of the atomic attributes, n is the number of atoms, and k is the number of nearest neighbors. We unfold the input along the n and k axis to obtain $X \in \mathbb{R}^{d\times n'}$, where $n' = n \times k$, and then the conventional scaled dot-product attention can be applied.

We prove the scaled dot-product attention satisfies the principle of permutation invariance in Section B. Because the atom-shared scaled dot-product attention is a permutation equivariance and the neighbor features $e_i^{(l)}$ are obtained by using the aggregate function ($\sum$) over all neighboring atoms, $e_i^{(l)}$ is a permutation invariance.

To highlight the atomic attributes of center atom, we extracted its features separately through the atom feature extraction subnetwork,

$$f_i^{(l)} = h_\Psi(x_i^{(l)}) \tag{6}$$

The overall network is deep due to the various operations for feature extraction. The alleviate the problem of vanishing gradient in deep neural networks, we adopted the shortcut connections (He et al., 2016), and the feature vector of point i at the layer $(l + 1)$ is updated by:

$$x_i^{(l+1)} = h_\Omega(e_i^{(l)}, f_i^{(l)}) + \text{Upsampling}(x_i^{(l)}) \tag{7}$$

In addition, different molecules consist of different numbers of atoms, and thus we need to align the atomic point sets into the input of a fixed size. The point clouds in computer vision applications are usually dense, and the subset points sampled from the cloud may serve as the input of the same size. For atomic point clouds, the sampling of atoms may miss significant information about the molecular structure. Here, the atomic points are padded into a fixed size (e.g., 300) with zeros, and a binary mask is applied on the point set at every layer, which guarantees that the output of MolAConv has the same number of points as the input. In the end, the elemental operation is sequentialized to form the encoder of the neural network for learning the molecular representation at different scales.

3.2 Alignment Network for Rotation and Translation

In order to satisfy the principle of transformation invariance, we designed an alignment mini-neural network called **TnRNet** (Translation and Rotation Network) to learn the transformation parameters from the input 3D coordinates, including the rotation angles $\alpha, \beta, \gamma \in [0, 2\pi)$ along x, y, z-axis and the translation shifts u, v, w along x, y, z-axis. Specifically,

$$(\alpha/2\pi \quad \beta/2\pi \quad \gamma/2\pi \quad u \quad v \quad w) = \text{Softmax}(h_\Phi(z_i^{(0)})) \tag{8}$$

where $z_i^{(0)}$ denotes the x, y, z-coordinates of atoms-i in layer-0, h_Φ is concretized as a shared-MLP. Then the transformation matrix M can be assembled as:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

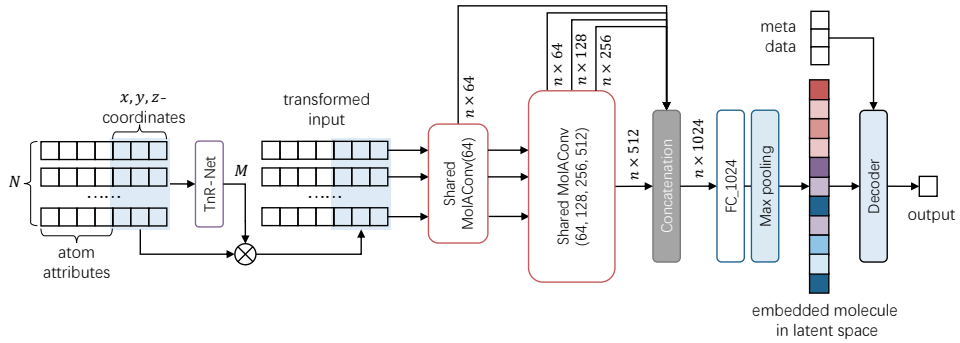

Figure 2: The architecture of Mol3DNet. Mol3DNet is a point-based neural network that uses the MolAConv as the elemental operation. The input of the network is the x, y, z-coordinates and the attributes of the atoms shaped in a $N \times F$ matrix. The additional input of meta-data (e.g., the adduct ions type of collision cross-section prediction) is used when necessary. The output of the network can be the properties of the molecule, e.g., the collision cross section, the water solubility, etc.

which can be applied on the original x, y, z-coordinates, mapping a molecule to the canonical space before feature extraction. This transformation guarantees that the same molecule with the atomic coordinates being rotated and/or translated are embedded into the same representation vector in the latent space by the encoder. Notably, we do not enforce the encoder to be dilation or shear invariant because the distances between all pairs of atomic points should remain the same for the same molecule because of the rigid chemical interactions between atoms.

The transformed coordinates can be processed by matrix M:

$$\begin{pmatrix} p'_x & p'_y & p'_z & 1 \end{pmatrix}^T = M \times \begin{pmatrix} p_x & p_y & p_z & 1 \end{pmatrix}^T \tag{10}$$

where p_x, p_y, p_z are the original coordinates, and p'_x, p'_y, p'_z are the transformed coordinates. The detailed architecture of TnRNet is illustrated in Figure 4 (a).

3.3 Architecture of Mol3DNet

Based on the elemental operation MolAConv and the alignment network TnRNet, we constructed the **Mol3Dnet**, a point-based deep neural network for molecular representation learning and molecular properties prediction from 3D conformers, as illustrated in Figure 2. The input of the network is an encoded 3D conformation (x, y, z-coordinates) and other related atomic attributes of a small molecule shaped a $N \times F$ matrix, where N denotes the maximum number of points, and F denotes the number of attributes (Table 6). The input coordinates are first transformed by the predicted transformation matrix, and the resulting point set is fed into the encoder established by with MolAConv. The features from different layers are concatenated and max-pooled into a vector in the latent space, i.e., the embedding vector. From the embedding vector, the output is obtained by a decoder consisting of five repeated residual fully connected blocks (details are in Section A.2). Also, the additional meta-data is used when the predicted properties come from different experimental conditions, such as the adduct ions types for the prediction of collision cross-section.

4 Experiments

The following settings for Mol3DNet are used in both classification and regression experiments. The maximum number of atoms in each molecule is limited to 300 (i.e., $n = 300$), which covers most of the chemical compounds of interest. The neighborhood number is set to 5 (i.e., $k = 5$) by default because it covers all covalent interactions and strong non-covalent interactions in molecules. We investigate the impact of the choice of k on the performance of Mol3DNet in Section E. The molecular conformers generated by ETKDG and OMEGA are taken (Section 4.3) as the input to Mol3DNet, whose results are reported separately. The detailed network configurations and hyper-parameters' values can be seen in Section D.

Category	Model	BBBP	Tox21	ToxCast	Sider	ClinTox	HIV
Graph-Based Methods	GCN	69.2	72.2	57.0	52.7	79.2	71.3
	GIN	63.0	72.4	58.6	52.3	78.0	69.5
	NFP	69.3	75.7	60.5	56.3	76.7	71.9
	GraphMVP	68.5	74.5	62.7	62.3	79.0	74.8
	GraphMVP-G	70.8	75.9	63.1	60.2	79.1	76.0
	GraphMVP-C	72.4	74.4	63.1	63.9	77.5	<u>77.0</u>
	SphereNet	70.7	<u>76.5</u>	<u>75.4</u>	65.8	80.9	71.8
Point-Based Methods	SchNet	67.3	72.7	63.5	58.8	79.5	76.8
	PointNet	68.6	70.5	74.4	63.3	78.2	76.9
	DGCNN	72.3	75.5	75.2	63.2	<u>81.5</u>	75.1
	Mol3DNet-ETKDG	<u>73.6</u>	76.1	**76.6**	<u>67.1</u>	**82.9**	76.9
	Mol3DNet-OMEGA	**75.9**	**76.6**	74.6	**69.1**	79.9	**77.4**

Table 2: Comparison of the molecular properties prediction (classification) by different models. The mean AUC-ROC scores are reported as the measurement. The best results are shown in bold and the second best results are shown with underlines.

4.1 Classification Tasks for Properties Prediction

Datasets. We set up the benchmark with seven classification datasets: *BBBP*, *Tox21*, *ToxCast*, *Sider*, *ClinTox*, and *HIV*, proposed by (Wu et al., 2018). Each dataset contains thousands of molecules SMILES strings labeled with binary classes (the exact numbers are shown in Table 7). Scaffold splitting is used to partition the molecules into a training and a testing subset in the ratio of 9:1.

Baselines. We compared Mol3DNet with a selected set of DNN models designed for MRL and chemical properties prediction, including the state-of-the-art graph-based and point-based methods. GCN (Kipf & Welling, 2016), GIN (Xu et al., 2018), and NFP (Duvenaud et al., 2015) are the classical graph-based DNN models[1], while GraphMVP (Jing et al., 2021) and SphereNet (Liu et al., 2021b) are recently developed graph-based DNN models[2]. GraphMVP, GraphMVP-C, and GraphMVP-G are pretrained on GEOM (Axelrod & Gomez-Bombarelli, 2022), containing 37 million energy-annotated conformations for over 450,000 molecules. For the point-based methods, we compared Mol3DNet against SchNet (Schütt et al., 2017), a classical model for MRL. We also implemented two baseline point-based models, PointNet (Qi et al., 2017) and DGCNN (Manessi et al., 2020), both equipped with the same decoder for MRL, which makes it a fair comparison against Mol3DNet. It is worth noting that in our implementations of these two models, in addition to the coordinates of the points, we incorporate the atomic attributes in the same way as the input to Mol3DNet so that these point-based models are adapted for molecules.

Results. The complete classification results are summarized in Table 2 (for detailed results see GitHub[3]). We observe that the baseline point-based methods (PointNet and DGCNN) perform better than the classical (GCN, GIN and NFP), and comparably with more recent graph-based methods (SphereNet and GraphMVP), which shows the advantage of representing molecules as point sets. On the other hand, our methods, Mol3DNet-ETKDG and Mol3DNet-OMEGA, performed better than the baseline and the classical (SchNet) point-based models, and achieved the highest AUC-ROC score in all tasks. It is worth noting that both Mol3DNet models were not pretrained on the GEOM dataset as the GraphMVP models (Jing et al., 2021). We anticipated the performance of our models will be further improved after pretraining. In addition, the attentional score maps are visualized in Section F.

4.2 Regression Tasks for Properties Prediction

Datasets. The following three recent experimental datasets are chosen to compare the performance of the models on the regression tasks for molecular properties prediction: *AllCCS* (Zhou et al., 2020) of molecules' collisional cross section (CCS) measured by ion mobility spectrometry (IMS), *SMRT* (Domingo-Almenara et al., 2019) of molecules' elution time (ET) in liquid chromatography

[1]GCN, GIN and NFP were implemented in the `TorchDrug` package (Tang et al., 2021)

[2]SphereNet was implemented in the `DIG` package (Liu et al., 2021a).

[3]They are temporarily placed in supplementary material, which will be public on GitHub soon.

Category	Model	AllCCS & BushCCS		SMRT		AqSolDB	
		MAE ↓	R2 ↑	MAE ↓	R2 ↑	MAE ↓	R2 ↑
Graph-Based Methods	GCN	19.593	0.550	104.867	0.574	1.261	0.511
	GIN	19.922	0.462	94.309	0.611	1.193	0.558
	NFP	22.354	0.577	84.619	0.697	1.153	0.614
	GraphMVP	11.243	0.823	54.495	0.817	0.719	0.792
	GraphMVP-G	9.006	0.860	53.511	**0.818**	0.714	0.798
	GraphMVP-C	9.226	0.855	54.508	**0.818**	0.719	0.793
Customized Methods	AllCCS	9.781	0.781	-	-	-	-
	SMRT-DLM	-	-	58.534	0.775	-	-
	AqSolPred	-	-	-	-	**0.559**	**0.872**
Point-Based Methods	SchNet	9.363	0.760	97.924	0.571	0.800	0.754
	PointNet	8.267	0.833	58.249	0.749	0.801	0.780
	DGCNN	8.128	0.826	52.506	0.788	0.864	0.740
	Mol3DNet-ETKDG	7.342	0.884	52.354	0.769	0.772	0.770
	Mol3DNet-OMEGA	**4.801**	**0.943**	**51.109**	0.809	0.746	0.770

Table 3: Comparison of deep learning models on the regression tasks for molecular properties prediction. The mean absolute error (MAE) and the coefficient of determination (R2) is used as the metrics. The best and the second best result is highlighted in bold and underlined respectively.

(LC), and *AqSolDB* (Sorkun et al., 2019) of molecules' solubility in water. Except for the CCS dataset, we randomly split the whole dataset into the training and test subsets by 9:1. For the comparison with the latest CCS prediction method (Zhou et al., 2020), which as trained on the whole AllCCS dataset but as not open-sourced, we trained all models using the entire ALLCCS dataset and then used another CCS dataset from Bush Lab (Bush et al., 2010) (*BushCCS*) that which does not overlap with AllCCS, as the testing set for all the models. The adduct ion types are encoded into one-hot encoding as the meta-data for CCS prediction. The statistics of all the datasets, including their sizes, ranges, mean, and standard deviation (SD), are summarized in Table 8.

Baselines. Besides the baselines used in the benchmark of the classification tasks, the customized machine learning models (Zhou et al., 2020; Domingo-Almenara et al., 2019; Sorkun et al., 2021) designed for the specific tasks are also included in the comparison.

Results. The point-based models (SchNet, PointNet, DGCNN, and Mol3DNet) perform better than the classical graph-based models (GCN, GIN, and NFP) in all three tasks. PointNet and DGCNN perform comparably with the pretrained graph-based models (GraphMVP) on the three tasks, while the two Mol3DNet models perform significantly better than GraphMPV models on CCS prediction, while Mol3DNet models perform slightly worse than GraphMPV models on the solubility prediction. It is worth noting that a molecule' CCS is highly dependent on the 3D conformations (Nielson et al., 2021), whereas the solubility is dependent mostly on the molecule's functional groups. As a result, it is not surprising to observe that the point-based models perform the best for the CCS prediction, but not so well on the solubility prediction. In particular, all deep learning models perform worse than the customized machine learning model (Sorkun et al., 2021)[4] using manually crafted features as input for solubility prediction, indicating MRL may not be necessary for all tasks of molecular properties prediction.

4.3 Ablation Study: Effects of Molecular Conformers

In this section, we compare the performance of different molecular conformer generation methods on the experimental conformers' dataset, and study their impact on properties prediction.

Datasets. We compare the popular computational molecular conformers generation algorithms on the Platinum Diverse Dataset (Friedrich et al., 2017), a high-quality benchmarking dataset of $2,859$ protein-bound ligand conformations extracted from the Protein Data Bank (Berman et al., 2000). For each molecule, only the conformers with the lowest energy are generated, which is the closest status

[4]We show different results here with Sorkun et al. (2021)'s paper because in their experiments the training set and test set are overlapped. We remove the overlapped subset from the training set and repeat the experiments.

	Cost/License	RMSD (Å)	Speed
2D	free/BSD	46.432	714.750
ETKDG	free/BSD	46.416	26.229
ETKDGv3	free/BSD	46.418	19.449
OMEGA	commercial	43.292	8.962

Table 4: Comparison of 3D conformers generation algorithms on the Platinum Diverse Dataset. The speed is measured by the number of molecules processed by each algorithm per second.

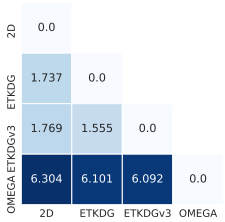

Figure 3: RMSD correlation matrix among molecular conformers.

to experimental benchmarks. Root-mean-square deviation (RMSD) using Kabsch algorithm (Kabsch, 1976) for rotation is used as the metrics.

Results. As Table 4 shows, OMEGA performs best in terms of accuracy. However, when generating the lowest energy conformer for each molecule, OMEGA does not show an obvious advantage in speed. The correlations of RMSDs between different conformers are plotted in Figure 3, which shows that the average RMSD between ETKDG and ETKDGv3 conformers is relatively small (1.555 Å). Therefore, in our experiments, we trained two Mol3DNet models using the input of the conformers generated by the open-source algorithm ETKDG, and the commercial algorithm OMEGA, respectively. The comparison of these two types of models are shown in Table 2 and Table 3. We observed that some tasks were sensitive to the conformers, for instance, BBBP, Sider, and CCS, in which Mol3DNet-OMEGA performs significantly better than Mol3DNet-ETKDG, probably because these molecular properties are highly dependent on 3D molecular conformations. However, in most cases, the ETKDG and OMEGA conformers achieve similar performances for molecular properties prediction. Hence, ETKDG is usable in most tasks as an open-source conformer generation algorithm.

4.4 Ablation Study: Effects of Components in MolAConv

The improvement of each components in MolAConv on *BBBP* classification is shown in Table 5. All the results are based on the ETKDG conformers. The other configurations are the same as the experimental setup in Section D. The MolAConv using atom shared-head attention, sum aggregate function, and skip connection perform best.

Components			AUC-ROC Score
Atom Shared Multi-Head Attention	Aggregate Function	Skip Connection	
✓	sum	✓	73.6
✓	max	✓	70.4
×	sum	✓	67.7
✓	sum	×	65.8

Table 5: Ablation study results of components in MolAConv.

5 Conclusion and Future Work

In this paper, we present Mol3DNet, a deep neural network for molecular representation learning that represents the molecules into point sets instead of molecular graphs, and follows two principles of point-based DNN (permutation invariance and transformation invariance, respectively). The experiments on regression and classification tasks demonstrate that Mol3DNet performs state-of-the-art except for solubility prediction. We anticipated that Mol3DNet could be enhanced by pretraining on GEOM dataset (Axelrod & Gomez-Bombarelli, 2022). In addition, our approach connected the molecules with point sets, so that point-based generative models may be adapted for molecular generative models.

REFERENCES

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):1–14, 2022.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Matthew F Bush, Zoe Hall, Kevin Giles, John Hoyes, Carol V Robinson, and Brandon T Ruotolo. Collision cross sections of proteins and their complexes: a calibration framework and database for gas-phase structural biology. *Analytical chemistry*, 82(22):9557–9565, 2010.

Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

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.

Xavier Domingo-Almenara, Carlos Guijas, Elizabeth Billings, J Rafael Montenegro-Burke, Winnie Uritboonthai, Aries E Aisporna, Emily Chen, H Paul Benton, and Gary Siuzdak. The metlin small molecule dataset for machine learning-based retention time prediction. *Nature communications*, 10(1):1–9, 2019.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

David Fooshee, Aaron Mood, Eugene Gutman, Mohammadamin Tavakoli, Gregor Urban, Frances Liu, Nancy Huynh, David Van Vranken, and Pierre Baldi. Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering*, 3(3):442–452, 2018.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

Nils-Ole Friedrich, Agnes Meyder, Christina de Bruyn Kops, Kai Sommer, Florian Flachsenberg, Matthias Rarey, and Johannes Kirchmair. High-quality dataset of protein-bound ligand conformations and its application to benchmarking conformer ensemble generators. *Journal of Chemical Information and Modeling*, 57(3):529–539, 2017.

Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

Francesco Gentile, Vibudh Agrawal, Michael Hsing, Anh-Tien Ton, Fuqiang Ban, Ulf Norinder, Martin E Gleave, and Artem Cherkasov. Deep docking: a deep learning platform for augmentation of structure based drug discovery. *ACS central science*, 6(6):939–949, 2020.

Paul CD Hawkins and Anthony Nicholls. Conformer generation with omega: learning from the data set and the analysis of failures. *Journal of chemical information and modeling*, 52(11): 2919–2936, 2012.

Paul CD Hawkins, A Geoffrey Skillman, Gregory L Warren, Benjamin A Ellingson, and Matthew T Stahl. Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of chemical information and modeling*, 50(4):572–584, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Baoyu Jing, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. Graph-mvp: Multi-view prototypical contrastive learning for multiplex graphs. *arXiv preprint arXiv:2109.03560*, 2021.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32 (5):922–923, 1976.

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

Johannes Klicpera, Shankari Giri, Johannes T Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.

Antonio Lavecchia. Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug discovery today*, 24(10):2017–2032, 2019.

Genyuan Li, Carey Rosenthal, and Herschel Rabitz. High dimensional model representations. *The Journal of Physical Chemistry A*, 105(33):7765–7777, 2001.

Yibo Li, Liangren Zhang, and Zhenming Liu. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):1–24, 2018.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, Keqiang Yan, Haoran Liu, Cong Fu, Bora M Oztekin, Xuan Zhang, and Shuiwang Ji. DIG: A turnkey library for diving into graph deep learning research. *Journal of Machine Learning Research*, 22(240):1–9, 2021a. URL http://jmlr.org/papers/v22/21-0343.html.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2021b.

Franco Manessi, Alessandro Rozza, and Mario Manzo. Dynamic graph convolutional networks. *Pattern Recognition*, 97:107000, 2020.

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

Felicity F Nielson, Sean M Colby, Dennis G Thomas, Ryan S Renslow, and Thomas O Metz. Exploring the impacts of conformer selection methods on ion mobility collision cross section predictions. *Analytical chemistry*, 93(8):3830–3838, 2021.

Giorgio Pesciullesi, Philippe Schwaller, Teodoro Laino, and Jean-Louis Reymond. Transfer learning enables the molecular transformer to predict regio-and stereoselective reactions on carbohydrates. *Nature communications*, 11(1):1–8, 2020.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574, 2015.

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

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

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.

Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. Aqsoldb, a curated reference set of aqueous solubility and 2d descriptors for a diverse set of compounds. *Scientific data*, 6(1):1–8, 2019.

Murat Cihan Sorkun, JM Vianney A Koelman, and Süleyman Er. Pushing the limits of solubility prediction via quality-oriented data selection. *Iscience*, 24(1):101961, 2021.

Jian Tang, Fei Wang, and Feixiong Cheng. Artificial intelligence for drug discovery. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 4074–4075, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hongwei Wang, Weijiang Li, Xiaomeng Jin, Kyunghyun Cho, Heng Ji, Jiawei Han, and Martin D Burke. Chemical-reaction-aware molecule representation learning. *arXiv preprint arXiv:2109.09888*, 2021a.

Kaili Wang, Renyi Zhou, Yaohang Li, and Min Li. Deepdtaf: a deep learning method to predict protein–ligand binding affinity. *Briefings in Bioinformatics*, 22(5):bbab072, 2021b.

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*, pp. 429–436, 2019.

Shuzhe Wang, Jagna Witek, Gregory A Landrum, and Sereina Riniker. Improving conformer generation for small rings and macrocycles based on distance geometry and experimental torsional-angle preferences. *Journal of chemical information and modeling*, 60(4):2044–2058, 2020.

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.

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.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Zhiwei Zhou, Mingdu Luo, Xi Chen, Yandong Yin, Xin Xiong, Ruohong Wang, and Zheng-Jiang Zhu. Ion mobility collision cross-section atlas for known and unknown metabolite annotation in untargeted metabolomics. *Nature communications*, 11(1):1–13, 2020.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

A ARCHITECTURAL DETAILS

A.1 ATOM FEATURE SPECIFICATION

In addition to x, y, z-coordinates, we chose 19 essential atomic attributions (Rogers & Hahn, 2010; Duvenaud et al., 2015; Coley et al., 2017) as the input features. All of the atomic attributions shown in Table 6 can be obtained by `RDKit` package. The molecules are removed if containing any atom not in the following types: `C`, `H`, `O`, `N`, `F`, `S`, `Cl`, `P`, `B`, `Br`, `I`.

Index	Description	Data Type
0-2	x, y, z coordinates	vector
3-14	atom type (`C`, `H`, `O`, `N`, `F`, `S`, `Cl`, `P`, `B`, `Br`, `I`)	one-hot encoding
15	number of immediate neighbors who are nonhydrogen atoms	scalar
16	valence minus the number of hydrogens	scalar
17	atomic mass	scalar
18	atomic charge	scalar
19	number of implicit hydrogens	scalar
20	is aromatic	boolean
21	is in a ring	boolean

Table 6: Encoding of the atomic attributes in the input to the 3D molecular DNN.

A.2 ARCHITECTURE OF TNRNET AND DECODER

We design TnRNet to predict the translation and rotation parameters from atomic x, y, z-coordinates. It is a mini network using shared-MLP as elemental operation, max pooling as the aggregate function, and sigmoid as activation function limiting the predicted rotation angles (α, β, and γ) to $[0, 2\pi]$ and limiting the predicted translation distance (p, q, and r) to $[0, 1]$. The detailed structure is shown in Figure 4 (a). Referring to ResNet (He et al., 2016), we implement a decoder with stacked fully connected layers and skip connections shown in Figure 4 (b). It is also be used in the baseline models, PointNet (Qi et al., 2017) and DGCNN (Manessi et al., 2020) so that we could except the effects of the decoder when comparing different molecular representation learning methods.

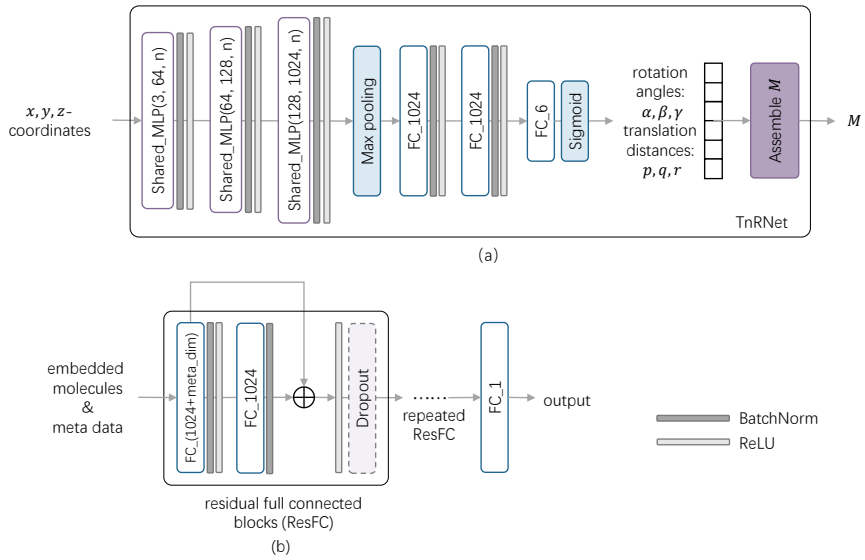

Figure 4: (a) Using TnRNet to establish the transform matrix M. In TnRNet, the x, y, z-coordinates of atoms are used to predict the rotation angles α, β, γ and translation distance p, q, r in axes. The transform matrix M is assembled in Equation 9. (b) The decoder consists of 5 residual fully connected blocks (ResFC). Each block is equipped with fully connected layers with skip connections. The last three ResFC is equipped with dropout to release the overfitting.

B PERMUTATION INVARIANCE PROVE

Definition B.1 (Permutation Equivariance)**.** Let π be a permutation of n elements. The permutation of $X \in \mathbb{R}^{d\times n'}$ can be represented as XP_π, where $P_\pi \in \mathbb{R}^{n'\times n'}$ denotes the permutation matrix associated with π, defined as $P_\pi = [e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(n)}]$, and e_i is a one-hot vector of length n with its i-th elements being 1. We call the operation $\mathcal{A} : \mathbb{R}^{d\times n'} \to \mathbb{R}^{d\times n'}$ a permutation equivariance if $\mathcal{A}(XP_\pi) = \mathcal{A}(X)P_\pi$.

Definition B.2 (Permutation Invariance)**.** The operation $\mathcal{A} : \mathbb{R}^{d\times n'} \to \mathbb{R}^{d\times n'}$ is a permutation invariance if $\mathcal{A}(XP_\pi) = \mathcal{A}(X)$.

Lemma B.1. If the input of the scaled dot-product attention is $X \in \mathbb{R}^{d\times n'}$, it is a permutation equivariance along the n' axis, i.e., Scaled Dot-Product Attention$(XP_\pi, XP_\pi, XP_\pi) =$ Scaled Dot-Product Attention$(X, X, X) \cdot P_\pi$.

Proof. We apply XP_π in Equation 3 and obtain,

$$\begin{aligned} \text{Left} &= XP_\pi \cdot \text{Softmax}((XP_\pi)^\mathsf{T} \cdot (XP_\pi)) \\ &= XP_\pi \cdot \text{Softmax}(P_\pi^\mathsf{T} \cdot X^\mathsf{T} \cdot X \cdot P_\pi) \\ &= XP_\pi P_\pi^\mathsf{T} \cdot \text{Softmax}(X^\mathsf{T} \cdot X)P_\pi \\ &= X \cdot \text{Softmax}(X^\mathsf{T} \cdot X)P_\pi = \text{Right} \end{aligned} \tag{11}$$

It should be noted that $P_\pi P_\pi^\mathsf{T} = I$ since P_π is an orthogonal matrix. And it is easy to verify that $\text{Softmax}(P_\pi^\mathsf{T} M P_\pi) = P_\pi^\mathsf{T}\text{Softmax}(M)P_\pi$, because each element in P_π is between 0 to 1. Therefore, we prove that the scaled dot-product attention is a permutation equivariance. □

C DATASETS OVERVIEW

The following Table 7 and Table 8 show the statistics information of the datasets used in the classification and regression experiments respectively. Comparing the molecular number before and after preprocess, it demonstrates that limiting the atom into 11 major types (`C`, `H`, `O`, `N`, `F`, `S`, `Cl`, `P`, `B`, `Br`, `I`) will only exclude a few molecules in the data set.

Dataset	Description	# Mol	# Mol after Prepossess	# Tasks
BBBP	Binary labels of blood-brain barrier penetration.	2039	2021	1
Tox21	Qualitative toxicity measurements on 12 biological targets, including nuclear receptors and stress response pathways.	7831	7615	12
ToxCast	Toxicology data based on in vitro high-throughput screening.	8575	7922	617
Sider	Marketed drugs and adverse drug reactions (ADR) dataset, grouped into 27 system organ classes.	1427	1313	27
ClinTox	Qualitative data of drugs approved by the FDA and those that have failed clinical trials for toxicity reasons.	1478	1454	2
HIV	Experimentally measured abilities to inhibit HIV replication.	41127	39289	1

Table 7: The statistics information of classification datasets.

D EXPERIMENTAL SETUP

The detailed configurations are shown in Table 9. All the generation algorithms of molecular conformers were executed on a server with 12 CPU (Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz) and GPU (GeForce RTX 2080 Ti). The 2D, ETKDG, and ETKDGv3 conformers were generated using the `RDKit` python package, and the OMEGA conformers were generated using the `OpenEye` python package.

Dataset	# Mol	# Mol after Prepossess	Range	Mean ± S.D.
AllCCS	3539	2193	[105.900, 322.500]	169.512±36.799
BushCCS	1224	1163	[108.800, 355.800]	174.066±34.501
SMRT	80038	79952	[0.300, 1471.700]	790.111±206.651
AqSolDB	9982	9041	[-13.171, 2.137]	-2.951±2.324

Table 8: The statistics information of regression datasets. Part of BushCCS is removed because they overlap with AllCCS data. In the experiments, we use AllCCS for training and BushCCS for testing, which causes data leaking if not removed the overlap data.

Hyperparameters	Values of classification	Values of regression
Batch size	16	16
Maximum atom number	300	300
Neighbors number (k)	5	5
Heads number (h)	4	4
Input dimension	21	21
Encoder layers size	64, 64, 128, 128, 256, 512, 1024	64, 64, 128, 128, 256, 512, 1024
Embedding dimension	2048	2048
Decoder layers size	2048, 1024, 512, 256, 128, 64	2048, 2048, 2048, 2048, 2048

Table 9: Values for hyper-parameters on all the regression and classification experiments.

E EXPERIMENTS WITH DIFFERENT NEIGHBORS NUMBER

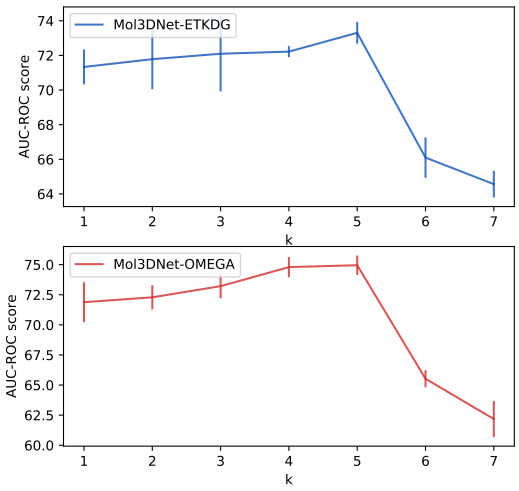

Figure 5: AUC-ROC scores on BBBP test subdataset of different number of neighbors.

Datasets. We use the *BBBP* dataset as an example to analyze the effect of number of neighbors.

Experiments configuration. We experimented the numbers of neighbors from 1 to 7 ($k = 1, 2, ..., 7$). To make a fair comparison, we fix the total number of atoms at 300 ($n = 300$), and the batch size at 16. For each selection of k, we repeat the experiments three times and report the mean and standard deviation of the AUC-ROC score.

Results. From Figure 5, when the number of neighbors is less than 5 ($k \leq 5$), the mean AUC-ROC score improves with the increasing k. However, when $k > 5$, the score decreases with the increasing k. This result suggests that for the BBBP dataset, it is sufficient to consider the interaction between an atom with five of its neighbors, and the interaction between more distant atoms has little or even a negative impact on improving the model performance.

F ATTENTION MAP VISUALIZATION

Based on the representation learning on the BBBP dataset using the same configuration as Section 4.1, the attention scores from different layers are visualized in Figure 6. To streamline the visualization, we plot the attention scores in two-dimensional molecular graphs. For the same central atom, the k-nearest neighbors are switched among the layers because of the dynamic feature extraction. The bottom MolAConv layers focus on the local structure (the neighbors are close to the central atom in Euclid Space) while the top layers focus on abstract global features.

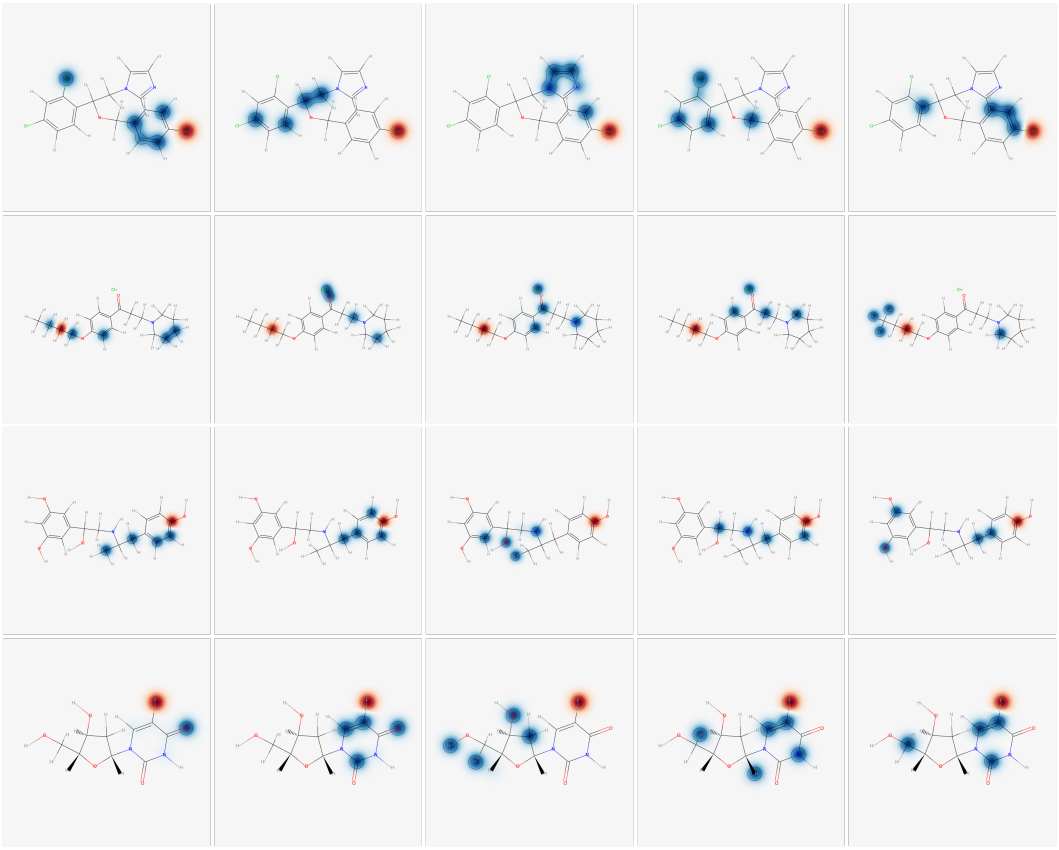

Figure 6: Heap maps of attention scores in each MolAConv layer. From left to right columns denote the 1st layer to the 5th layer, respectively. The center atom is labeled in red, and its k-nearest neighbors are labeled in blue. The higher the attention weight the darker the blue.