Chemical Engineering Journal 414 (2021) 128817

Contents lists available at ScienceDirect

Chemical Engineering Journal

journal homepage: www.elsevier.com/locate/cej

# Introducing block design in graph neural networks for molecular properties prediction

Yuquan Li [a], Pengyong Li [b], Xing Yang [a], Chang-Yu Hsieh [c], Shengyu Zhang [c], Xiaorui Wang [a], Ruiqiang Lu [a], Huanxiang Liu [d], Xiaojun Yao [a,*]

[a] College of Chemistry and Chemical Engineering, Lanzhou University, 730000, China
[b] Department of Biomedical Engineering, Tsinghua University, 100084, China
[c] Tencent Quantum Laboratory, Tencent, Shenzhen 518057, China
[d] School of Pharmacy, Lanzhou University, 730000, China

ARTICLE INFO

*Keywords:*
Block-based neural networks
Message passing
Molecular properties prediction

ABSTRACT

The number of states required for describing a many-body quantum system increases exponentially with the number of particles; thus, it is time- and effort-consuming to exactly calculate molecular properties. Herein, we propose a deep learning algorithm named block-based graph neural network (BGNN) as an approximate solution. The algorithm can be understood as a representation learning process to extract useful interactions between a target atom and its neighboring atomic groups. Compared to other graph model variants, BGNN achieved the smallest mean absolute errors in most tasks on two large molecular datasets, QM9 and Alchemy. Our advanced machine learning method exhibits general applicability and can be readily employed for bioactivity prediction and other tasks relevant to drug discovery and materials design.

## 1. Introduction

Quantum mechanics provides a rigorous description of the forces that control the behavior of atomic and molecular crystalline materials [1], and molecular properties calculation is important for drug discovery and material design. In the past, molecular properties are usually calculated by solving the Schrödinger equation [2], which consumes much time and effort. When dealing with large-scale molecular systems such as biomolecules, methods based on molecular mechanics (MM) [3,4] offer an appropriate tradeoff between accuracy and efficiency under various conditions. However, the MM methods have seen their limitation on accuracy in a wide range of scenarios [5].

There has been an ongoing endeavor to design more accurate algorithms for estimating molecular properties. In particular, machine learning (ML) has proven to be a useful tool for predicting molecular properties [6–11]. However, most ML models strongly rely on expert-crafted features, such as fingerprints and molecular descriptors. As the number of molecular data has been increasing enormously in this era of big data, the traditional machine learning methods, without a feature-learning capability, will gradually lose their advantages [12] as witnessed in other application domains of ML, such as computer visions [13–15] and natural language processing [16,17].

The rise of deep learning leads to many advances in predicting molecular properties [1,18–21] with the data-driven approach. With the development of various training techniques [22,23], deep learning methods have caught up with traditional ML methods in performance. At the same time, it enjoys some advantages such as high accuracy and automatic learning of the features.

Graph neural networks (GNNs) are deep learning based methods that operate on the graph domain [24,25]. When applied to molecular properties prediction, the graph neural networks take the mere molecule graph as input, with almost no feature engineering required. Despite this simplicity, GNNs have exhibited performance significantly better than previous machine learning models [6–11], showing the power of exploiting the natural graph structure in molecules.

In the beginning, the graph neural networks are used to generate molecular fingerprints [26] as the input of machine learning methods. It is gradually realized that graph networks can directly perform representation learning on molecules. Kearnes et al. [27] proposed to use molecular graph convolutions to predict molecular properties for high-throughput molecular screening. Schütt et al. [28] developed a deep tensor neural network that enables spatially and chemically resolved

* Corresponding author.
*E-mail address:* xjyao@lzu.edu.cn (X. Yao).

https://doi.org/10.1016/j.cej.2021.128817
Received 17 June 2020; Received in revised form 8 December 2020; Accepted 30 January 2021
Available online 5 February 2021
1385-8947/© 2021 Elsevier B.V. All rights reserved.

**Table 1**
The property descriptions and units about the prediction tasks into QM9 and Alchemy dataset.

| Task | Property description | Unit |
|---|---|---|
| mu | Dipole moment | *Debye* |
| Alpha | Polarizability | $a_0^3$ |
| HOMO | Highest occupied molecular orbital | $E_h$ |
| LUMO | Lowest unoccupied molecular orbital | $E_h$ |
| gap | HOMO-LUMO gap | $E_h$ |
| R2 | Electronic spatial extent | $a_0^2$ |
| zpve | Zero point vibrational energy | $E_h$ |
| U0 | Internal energy at 0 K | $E_h$ |
| U | Internal energy at 298.15 K | $E_h$ |
| H | Enthalpy at 298.15 K | $E_h$ |
| G | Free energy at 298.15 K | $E_h$ |
| Cv | Heat capacity at 298.15 K | $E_hK^{-1}$ |

insights into quantum-mechanical observables of molecular systems. Gilmer et al. [12] summarized these models and reinterpreted them using message passing neural networks. Since then, some other variants of message passing neural networks have been proposed [1,29–34]. Even though graph neural networks manifest many encouraging results, their performance is actually limited by the network degradation problem without a universal solution [35]. Network degradation means that a network's performance gets saturated and followed by rapid degradation as the network depth increases.

In this work, we propose a new graph learning paradigm based on a block design named block-based graph neural network (BGNN) and demonstrate that the network degradation problem can be reduced by applying block design with normalization and skip-connection. This learning paradigm enables more efficient information flow in the network due to the layer-by-layer representation learning. We then test BGNN on QM9 [38] and Alchemy [39], two largest benchmark datasets for molecular properties. As shown by experimental results, our BGNN can accurately predict molecular properties. Compared to the previous works, the estimation mean absolute error on the test set is more acceptable for practice.

## 2. Material and methods

### 2.1. Dataset

#### 2.1.1. QM9 dataset

We select the QM9 [38] dataset, which is commonly used as a benchmark to validate graph neural networks designed for molecular sciences. The dataset can be downloaded from MoleculeNet [40], which contains the structural data and thirteen molecular properties for roughly 130 k small molecules with up to nine non-hydrogen heavy atoms made of C, O, N, and F. All molecular geometries were relaxed, and their properties were calculated at the DFT B3LYP/6-31G(2df,p) level of theory. Here, we use twelve out of thirteen properties for the prediction task, as shown in Table 1. The molecules are randomly split to a training set, a validation set, and a test set with an 8:1:1 ratio.

#### 2.1.2. Alchemy dataset

Alchemy [39] is one of the latest and largest molecular property benchmark dataset. It contains 130 k organic molecules with the same twelve molecular properties as of QM9. Those molecules were selected from the GDB MedChem [41] database, which has presumably better pharmaceutical relevance than MoleculeNet. Each molecule in Alchemy dataset has 9–12 non-hydrogen atoms, made up of C, N, O, F, S, and Cl [41]. These molecular properties are calculated using the Python-based Simulations of Chemistry Framework (PySCF) [42]. Here, we consider all twelve properties, as shown in Table 1. The dataset is also randomly split into a training set, a validation set, and a test set in the ratio of 8:1:1. We downloaded the 130 k version dataset used in Chen et al. [39] from Tencent Quantum Laboratory.

**Table 2**
Atom features and bond features used in the molecular graph.

| Type | Feature | Description | Size |
|---|---|---|---|
| Node feature | Atom type | H, C, N, O, F, S, Cl (one-hot) | 7 |
| | Atomic number | Atomic number of the atom (integer) | 1 |
| | Acceptor | Whether or not to accepts electrons (binary) | 2 |
| | Donor | Whether or not to donates electrons (binary) | 2 |
| | Aromatic | In an aromatic system (binary) | 2 |
| | Hybridization | sp, $sp^2$, $sp^3$ (one-hot or null) | 3 |
| | Num. Hs. | Number of hydrogens (integer) | 1 |
| | Explicit valence | Explicit valence of this atom (integer) | 1 |
| | Implicit valence | Implicit valence of this atom (integer) | 1 |
| | Formal charge | Formal charge of this atom (integer) | 1 |
| | Num. explicit Hs. | Number of implicit Hs. this atom is bound to (integer) | 1 |
| | Num. radical electrons | Number of radical electrons for this atom (integer) | 1 |
| Edge feature | Bond type | Single, double, triple, aromatic (one-hot) | 4 |
| | Distance | Euclidean distance between two atoms (float) | 1 |
| | Conjugated | Whether the bond is conjugate (binary) | 2 |

#### 2.1.3. Dataset preprocessing

To enable graph-based learning, we process all molecules into molecular graphs with basic features, as shown in Table 2. In practice, molecular graphs are often transformed into matrix representations to save memory and improve calculating efficiency [43], including node features, edge features, and atomic coordinates, as shown in Fig. 1. The atom feature matrix contains all atoms and their feature information. The bond feature matrix contains all bonds and bond features. The coordinates matrix contains the coordinate information of the atoms.

Considering the long-distance propagation, we added virtual bonds to the graph to convert a molecular graph into a fully connected graph, as shown in Fig. 2, which is more conducive to message passing [12] and allows the message to pass over long distances. A fully connected graph means that all the atom pairs are connected, even if there are no bonds between them.

### 2.2. Block-based graph neural networks

#### 2.2.1. Network degradation

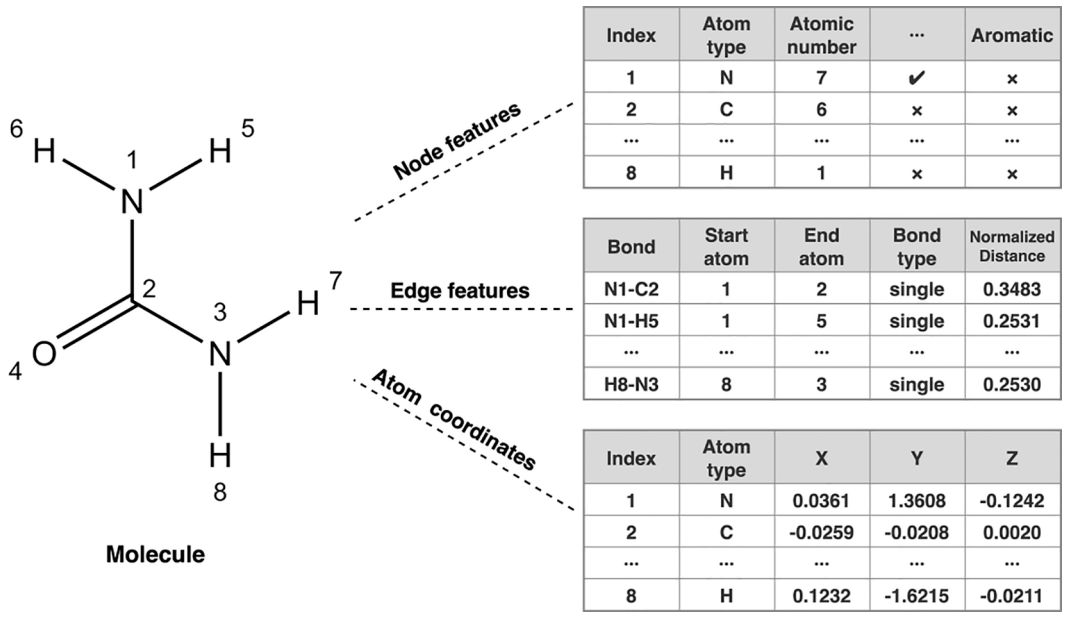
Following the intuition of layer-by-layer feature extraction, researchers tend to design deeper networks with a more remarkable ability for representation learning [44]. However, a very general degradation problem has been exposed during the training of deeper networks: as the network depth increases, accuracy gets saturated and followed by a rapid degradation [14]. Thus, not all graph neural networks can be easily optimized.

Li et al. [35] proposed to add skip connections and dilated convolutions, which can deepen the Laplacian based graph neural networks to 56 layers without performance degradation, indicating that network degradation can be mitigated with skip connections.

#### 2.2.2. Block-based GNN design

The complexity of layer-by-layer stacking makes it challenging to design new network architectures. Herein, we propose to build graph neural networks using block design. The block design for neural network was first developed in Simonyan et al. [13] by VGG (Visual Geometry Group). Since then, many landmark models based on block design have been created [14–16,45].

A graph neural network usually consists of the following layers:




| Index | Atom type | Atomic number | ... | Aromatic |
|---|---|---|---|---|
| 1 | N | 7 | ✔ | × |
| 2 | C | 6 | × | × |
| ... | ... | ... | ... | ... |
| 8 | H | 1 | × | × |

| Bond | Start atom | End atom | Bond type | Normalized Distance |
|---|---|---|---|---|
| N1-C2 | 1 | 2 | single | 0.3483 |
| N1-H5 | 1 | 5 | single | 0.2531 |
| ... | ... | ... | ... | ... |
| H8-N3 | 8 | 3 | single | 0.2530 |

| Index | Atom type | X | Y | Z |
|---|---|---|---|---|
| 1 | N | 0.0361 | 1.3608 | -0.1242 |
| 2 | C | -0.0259 | -0.0208 | 0.0020 |
| ... | ... | ... | ... | ... |
| 8 | H | 0.1232 | -1.6215 | -0.0211 |

**Fig. 1.** Molecular graphs stored as atomic feature matrix, bond feature matrix, and atomic coordinate matrix in computers.

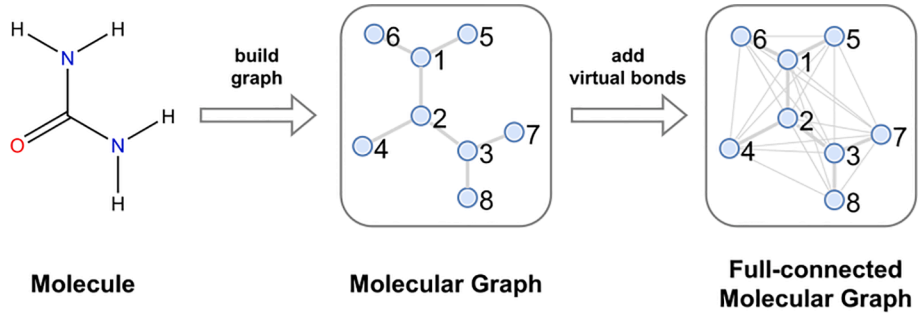



**Fig. 2.** The building of a fully connected molecular graph.

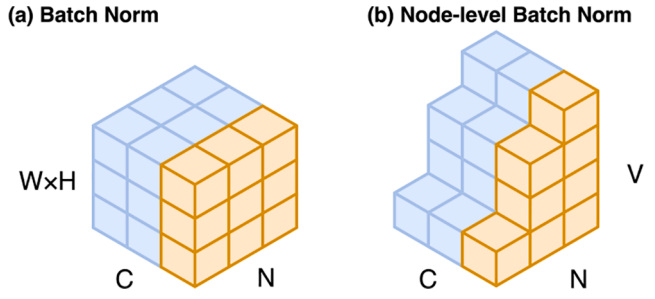



**Fig. 3.** From batch normalization to node-level batch normalization. (a) Represents a general batch normalization technique, where C refers to the feature channel, N refers to the number of samples, and W and H refer to an image's width and height. (b) Represents a node-level batch normalization technique, where C refers to an atom's feature channel, N refers to the number of molecular graphs, and V refers to the number of atomic vertices.

1. Message passing layer, or graph convolutional layer
2. Full-connected layer, or dense layer
3. Graph pooling layer
4. Graph normalization layer
5. Activation layer
6. Other mathematical operation layers

Block design selects and combines several layers above to build a block. Building graph neural networks through block design bring many conveniences.

### 2.2.3. *Normalization in block*

A fundamental assumption that most learning algorithms rely on is that the training and test samples are independent and follow the same distribution. However, inside a deep neural network, each layer has input affected by randomness in parameter initialization and input, causing the internal covariate shift problem [22] that hurts the efficiency and converge of the model's training.

Batch normalization (BN) is a widely-used method to mitigate this issue. During the training, the method normalizes the input to each layer to the same mean and variance. It allows the input value of the activation function to fall into a space that is more sensitive to the input, thereby avoiding the vanishing gradient problem.

We propose to use node-level batch normalization [26] for graph data, as shown in Fig. 3. The graph network has been around for a long time, but there are very few works [26] about normalization. Because the graph data does not have the concept of width and height, and the number of atoms is uncertain, batch normalization cannot be directly applied to molecular graph data. By generalizing the standard batch normalization to node-level batch normalization, we can normalize the feature of each node, making it zero mean and unit variance to solve the internal covariate shift problem.

#### 2.2.4. Mathematical description

For simplicity, we describe a block consists of several layers, which is applied to molecular graphs $G$ with node set $\{x_i : i = 1, \cdots, n\}$ and edge set $\{e_{ij} : i.j = 1, \cdots, n\}$, where $n$ is the number of atoms in the molecule. The forward propagation of a block-based graph neural network includes three phases: the graph learning phase, the transition operations phase, and the residual connection phase.

*2.2.4.1. First phase: message passing phase.* Let $h_i$ be the hidden state at each node in the message passing phase. We use the message passing process to describe graph networks. Each message passing process consists of two parts: message calculation and hidden state updating.

The message $m_{ji}$ for passing is calculated according to Eq. (1), where $M$ is the message function, and $N(i)$ is the neighbors of the node $i$ in graph $G$.

$$m_{ji} = \sum_{j \in N(i)} M\left(h_i, h_j, e_{ij}\right) \tag{1}$$

The message $m_{ji}$ can also be calculated by directed message passing [34,36,37]. It considers directed messages from neighboring atom pairs. The directed message passing calculates the directed message $m_{ji}^{directed}$ for updating according to Eqs. (1) and (2).

$$m_{ji}^{directed} = \sum_{k \in N(j)} M\left(m_{ji}, m_{kj}, e_{kj}^{*}\right) \tag{2}$$

In our works, we calculate the directed messages $m_{ji}^{directed}$ according to Eqs. (3) and (4).

$$m_{ji} = h_i + \sum_{j \in N(i)} h_j \cdot f_h\left(e_{ij}\right) \tag{3}$$

$$m_{ji}^{directed} = m_{ji} + \sum_{k \in N(j)} m_{kj} \cdot f_m\left(e_{kj}^{*}\right) \tag{4}$$

where $f_h$ is the edge $e_{ij}$ embedding function for the hidden state $h_j$, and $f_m$ is the edge $e_{kj}^{*}$ embedding function for the message $m_{kj}$. In this case, $N(j)$ does not include node $i$. In Eq. (4), $e_{kj}^{*}$ can include more information such as the angle of three nodes $i, j, k$ where $i$ is regarded as a constant node. Then, hidden states $h_i^{'}$ at each node in the graph are updated by a gated recurrent unit *GRU* according to Eq. (5).

$$h_i^{'} = GRU\left(h_i, m_{ji}^{directed}\right) \tag{5}$$

This description gives a complete message passing process. The entire process run recurrently $T$ times. Consequently, the hidden state $h_i^t$ will be updated $T$ times, and the hidden state $h_i^T$ is obtained at the end.

*2.2.4.2. Second phase: transition operations.* This stage is a set $O$ composed of some transition operations layers according to Eq. (6). The residual $x_i^{res}$ of node set $x_i^b$ is calculated in this phase. The set $O$ contains at least two layers: the node-level normalization layer and the activation function layer, in our case node-level batch normalization and ReLU.

$$x_i^{res} = O(h_i^T). \tag{6}$$

*2.2.4.3. Third phase: skip connection.* This stage adds the residual $x_i^{res}$ to the input node set $x_i^b$ according to Eq. (7).

$$x_i^{b+1} = x_i^{res} + x_i^b \tag{7}$$

The skip connection can also be an extension of residual connection, such as the dense connection [15]. The node set $x_i^b$ is updated in each neural network block, and the final node set $x_i^B$ is obtained.

Finally, the graph readout function $R$ is applied in the output stage. It can map the output graph with node set $x_i^B$ into molecular properties according to Eq. (8).

$$\hat{y} = R\left(\left\{x_i^B | i \in G\right\}\right) \tag{8}$$

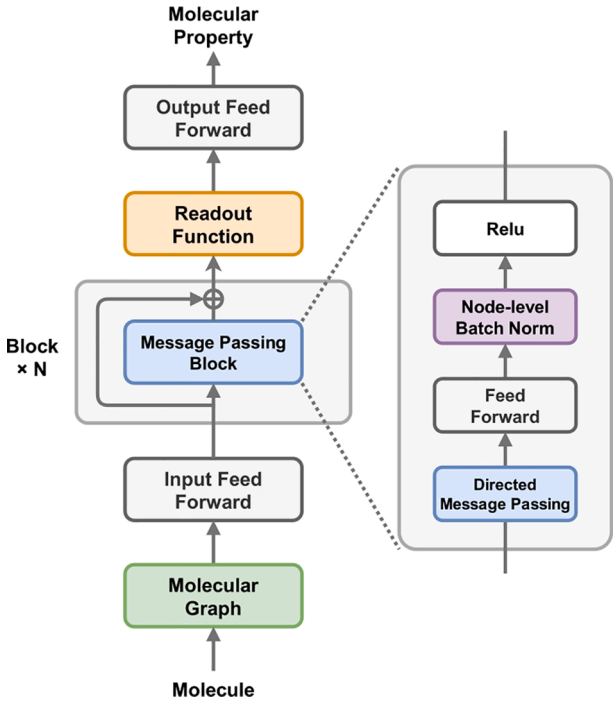



**Fig. 4.** The architecture of block-based graph neural network.

**Table 3**
The hyperparameters used in model training.

| Hyperparameter | Value |
|---|---|
| Epoch | 400 |
| Batch size | 64 |
| Optimizer | Adam [23] |
| Initial learning rate | 0.0001 |
| LR reduce patience | 5 |
| Early-stop patience | 30 |

### 2.3. Block-based graph neural network

Based on the description above, we design a neural network, as shown in Fig. 4. A molecule is first processed into a molecular graph as input. First, it is fed into a feed-forward neural network. The output is then processed by N residual blocks. Each block contains a directed message passing layer, a feed-forward neural network, and a node-level normalization layer. The readout function is used to map the output molecular graph into a flatten vector. This vector is finally mapped to a float number (corresponding to the predicted property) by a feed-forward neural network. As shown in Table 6, with the depth increases, the performance of BGNN will improve. We observe that the model performs best with five message passing blocks, so we choose BGNN with a depth of five as the experimental model.

We implemented the above model using pytorch_geometric [43], with the hyperparameters set shown in Table 3. We use a gated recurrent unit [46] neural networks for hidden status updates. Our readout function is Set2Set [47], which can capture more information than the global summing pool. The feed-forward neural network plays a role in enhancing the network's expressive ability, and the number of which usually ranges from one to three. To prevent overfitting and save time, we adopt the early-stop technique.

**Table 4**
Prediction mean absolute error on the QM9 datasets.

| Task | ECFP | CM | DTNN | MPNN | AttentiveFP | BGNN |
|---|---|---|---|---|---|---|
| mu | 0.602 | 0.519 | 0.244 | 0.358 | 0.451 | **0.057** |
| Alpha | 3.10 | 0.85 | 0.95 | 0.89 | 0.492 | **0.217** |
| HOMO | 0.0066 | 0.00506 | 0.00388 | 0.00541 | 0.00358 | **0.00225** |
| LUMO | 0.00854 | 0.00645 | 0.00513 | 0.00623 | 0.00415 | **0.00193** |
| gap | 0.01 | 0.0086 | 0.0066 | 0.0082 | 0.00528 | **0.00334** |
| R2 | 125.7 | 46 | 17 | 28.5 | 26.839 | **3.017** |
| zpve | 0.01109 | 0.00207 | 0.00172 | 0.00216 | 0.00120 | **0.00027** |
| U0 | 15.1 | 2.27 | 2.43 | 2.05 | 0.898 | **0.087** |
| U | 15.1 | 2.27 | 2.43 | 2.00 | 0.893 | **0.085** |
| H | 15.1 | 2.27 | 2.43 | 2.02 | 0.893 | **0.113** |
| G | 15.1 | 2.27 | 2.43 | 2.02 | 0.893 | **0.133** |
| Cv | 1.77 | 0.39 | 0.27 | 0.42 | 0.252 | **0.105** |

**Table 5**
Prediction mean absolute error on the 130 k version Alchemy dataset.

| Task | MPNN[a] | MPNN[b] | BGNN |
|---|---|---|---|
| mu | 0.0303 | 0.0210 | **0.00892** |
| Alpha | 0.0647 | 0.0459 | **0.0362** |
| HOMO | 0.0940 | 0.0733 | **0.0604** |
| LUMO | 0.0272 | 0.0163 | **0.00104** |
| gap | 0.1220 | **0.0928** | 0.0967 |
| R2 | 0.0272 | 0.0163 | **0.00324** |
| zpve | 0.0938 | 0.0564 | **0.0321** |
| U0 | 0.0272 | 0.0163 | **0.00111** |
| U | 0.0272 | 0.0163 | **0.00212** |
| H | 0.1023 | 0.0776 | **0.0562** |
| G | 0.1282 | 0.0819 | **0.0676** |
| Cv | 0.0425 | 0.0277 | **0.0174** |

[a] The best score with a random split from the original paper.
[b] The best score with a stratified split from the original paper.

**Table 6**
Prediction mean absolute error on QM9 dataset with different depth.

| Depth | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|
| mu | 0.075 | 0.070 | 0.070 | **0.057** | 0.081 |
| Alpha | 0.368 | 0.286 | 0.274 | **0.217** | 0.325 |
| HOMO | 0.00227 | 0.00216 | **0.00207** | 0.00225 | 0.00337 |
| LUMO | 0.00222 | 0.00202 | 0.00199 | **0.00193** | 0.00269 |
| gap | 0.00327 | **0.00318** | 0.00328 | 0.00334 | 0.00354 |
| R2 | 4.467 | 3.201 | 3.976 | 3.017 | **2.554** |
| zpve | 0.00026 | **0.00022** | 0.00027 | 0.00027 | 0.00032 |
| U0 | 0.121 | 0.105 | 0.091 | **0.087** | 0.186 |
| U | 0.129 | 0.128 | 0.112 | **0.085** | 0.134 |
| H | 0.117 | 0.094 | **0.091** | 0.113 | 0.111 |
| G | 0.132 | 0.114 | **0.109** | 0.133 | 0.165 |
| Cv | 0.139 | 0.123 | 0.121 | **0.105** | 0.119 |

Bold text indicates that this model obtains the smallest test set error.

## 3. Results

In this work, we test our algorithm on QM9 and Alchemy datasets. To validate models' prediction performance, we calculate the mean absolute error between the predicted value and the true value, the same as previous works [1,29–34]. The average of five repeated results is used as the final value. The depth of the BGNN model is five.

We compare the mean absolute error of our BGNN method to other state-of-the-art ones (ECFP, CM, DTNN, MPNN, and AttentiveFP) in previous work [33,40]) in Table 4. The prediction mean absolute error of ECFP, CM, DTNN, MPNN, and AttentiveFP are taken from the reference [33]. As can be seen, our method achieves the smallest mean absolute error in all the twelve molecular property prediction tasks.

From Table 5, eleven out of twelve prediction tasks got the smallest mean absolute errors compared to other methods. These further demonstrate the accuracy of the BGNN method as a robust learning algorithm for molecular properties.

## 4. Discussions

### 4.1. Impact of depth

To further analyze the performance of BGNN, we investigate the impact of the depth, that is, the message passing blocks number on the performance of BGNN, as shown in Table 6. As the depth increases, the performance of BGNN will improve. Even when the depth increases to 10 blocks, there are still performance improvements in some prediction tasks. These results indicate that deeper BGNN has a more remarkable ability to learn molecular representation. As shown in Fig. 5, we test four different models in six different depths. We found that a graph model will have an increase in prediction error as the depth deepens. When normalization and residual connection are added, the model will not show a significant performance degradation as the depth deepens. BGNN can reduce the impact of the network degradation problem and improve performance.

### 4.2. Impact of block design

We introduced block design to reduce the impact of network degradation and boost performance. Here, we selected a few representative tasks based on the test in Table 6, and investigate the impact of key layers in this block on performance by removing or keeping these layers. As shown in Fig. 5, we test four different models, and BGNN got the smallest mean absolute error in most cases. As the batch normalization and residual connection are removed, the prediction error is similar to MPNN. After adding batch normalization and residual connection, the model's prediction error has decreased on most prediction tasks, among which batch normalization brings a greater improvement than residual connection. The results indicate that block design with normalization and skip connection plays essential roles in our model.

## 5. Conclusions

In this paper, we proposed a new algorithm BGNN to predict molecular properties. The method introduced block design in graph neural networks with normalization and skip connection. When applied to molecule properties prediction, the method takes molecular graphs with basic features as inputs. We validate our algorithm on two large molecular properties datasets for benchmarking. As a result, BGNN achieved the smallest mean absolute errors in most tasks on two large molecular datasets. These experiments demonstrate the accuracy of the new method in predicting molecular properties.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

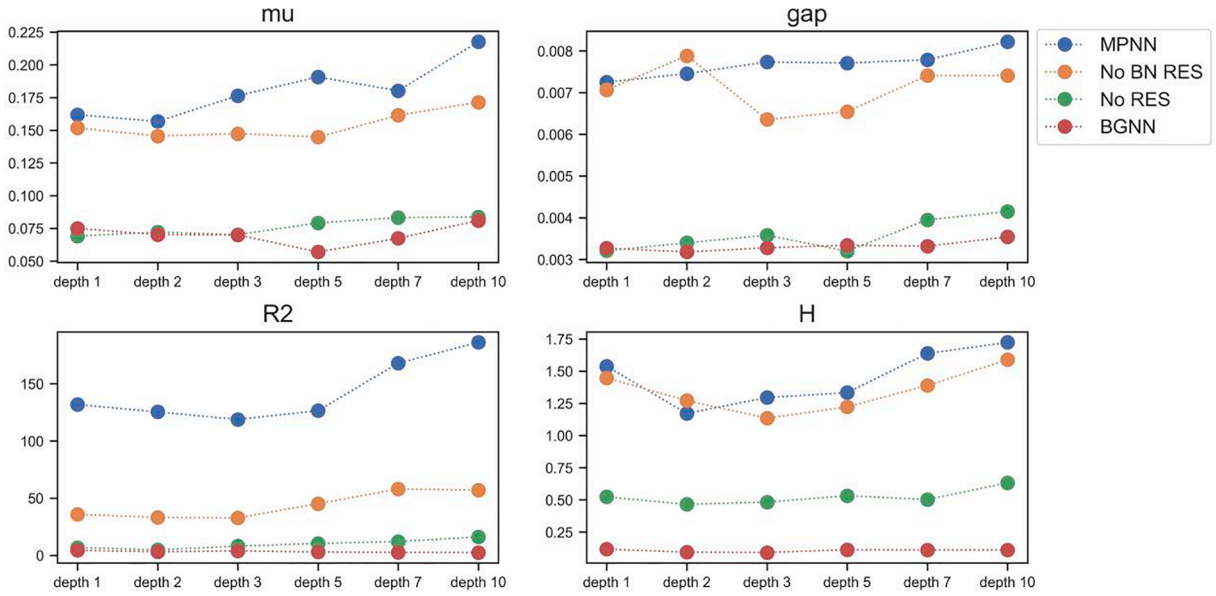



**Fig. 5.** Prediction MAE comparison of different models in different depths. Four models were used for validation. Model 1 is an MPNN using our features. Model 2 is BGNN with batch normalization and residual connection removed. Model 3 is BGNN with only residual connection removed. Model 4 is BGNN. The four models share the same hyperparameters, including the same depth, hidden nodes, epochs, learning rate, batch size, etc.

*Acknowledgment*

This work was supported by the National Natural Science Foundation of China (21775060).

*Dataset and code available*

The datasets are available at http://moleculenet.ai/datasets-1 and https://alchemy.tencent.com for QM9 and Alchemy. The code is available at https://github.com/yvquanli/bgnn.

## References

[1] A.V. Shapeev, Moment tensor potentials: a class of systematically improvable interatomic potentials, Multiscale Model. Simul. 14 (2016) 1153–1173, https://doi.org/10.1137/15M1054183.
[2] P. Hohenberg, W. Kohn, Inhomogeneous electron gas, Phys. Rev. 136 (1964) B864–B871, https://doi.org/10.1103/PhysRev.136.B864.
[3] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, J. Comput. Chem. 4 (1983) 187–217, https://doi.org/10.1002/jcc.540040211.
[4] B.R. Brooks, C.L. Brooks III, A.D. Mackerell Jr., L. Nilsson, R.J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R.M. Venable, H.L. Woodcock, X. Wu, W. Yang, D.M. York, M. Karplus, CHARMM: the biomolecular simulation program, J. Comput. Chem. 30 (2009) 1545–1614, https://doi.org/10.1002/jcc.21287.
[5] J.C.A. Boeyens, P. Comba, Molecular mechanics: theoretical basis, rules, scope and limits, Coord. Chem. Rev. 212 (2001) 3–10, https://doi.org/10.1016/s0010-8545(00)00353-2.
[6] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. 98 (2007) 146401, https://doi.org/10.1103/PhysRevLett.98.146401.
[7] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons, Phys. Rev. Lett. 104 (2010) 136403, https://doi.org/10.1103/PhysRevLett.104.136403.
[8] M. Rupp, A. Tkatchenko, K.-R. Müller, O.A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning, Phys. Rev. Lett. 108 (2012) 058301, https://doi.org/10.1103/PhysRevLett.108.058301.
[9] A.A. Lee, Q. Yang, V. Sresht, P. Bolgar, X. Hou, J.L. Klug-McLeod, C.R. Butler, Molecular transformer unifies reaction prediction and retrosynthesis across pharma chemical space, Chem. Commun. 55 (2019) 12152–12155, https://doi.org/10.1039/C9CC05122H.
[10] S. Chmiela, A. Tkatchenko, H.E. Sauceda, I. Poltavsky, K.T. Schütt, K.R. Müller, Machine learning of accurate energy-conserving molecular force fields, Sci. Adv. 3 (2017) e1603015, https://doi.org/10.1126/sciadv.1603015.
[11] M. Hirn, S. Mallat, N. Poilvert, Wavelet scattering regression of quantum chemical energies, Multiscale Model. Simul. 15 (2017) 827–863, https://doi.org/10.1137/16M1075454.
[12] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: Int. Conf. Mach. Learn., 2017, pp. 2053–2070.
[13] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Int. Conf. Learn. Represent. (2015).
[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Comput. Vis. Pattern Recognit., 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[15] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Comput. Vis. Pattern Recognit., 2017, pp. 2261–2269, https://doi.org/10.1109/CVPR.2017.243.
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Adv. Neural Inf. Process. Syst., 2017, pp. 5999–6009.
[17] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: North Am. Chapter Assoc. Comput. Linguist., 2019, pp. 4171–4186.
[18] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.R. Müller, O. Anatole Von Lilienfeld, Machine learning of molecular electronic properties in chemical compound space, New J. Phys. 15 (2013) 095003, https://doi.org/10.1088/1367-2630/15/9/095003.
[19] L. Zhang, J. Han, H. Wang, R. Car, E. Weinan, Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics, Phys. Rev. Lett. 120 (2018) 1–22, https://doi.org/10.1103/PhysRevLett.120.143001.
[20] J.S. Smith, O. Isayev, A.E. Roitberg, ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost, Chem. Sci. 8 (2017) 3192–3203, https://doi.org/10.1039/C6SC05720A.
[21] K. Yao, J.E. Herr, D. Toth, R. Mckintyre, J. Parkhill, The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics, Chem. Sci. 9 (2018) 2261–2269, https://doi.org/10.1039/C7SC04934J.
[22] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Int. Conf. Mach. Learn., 2015, pp. 448–456.
[23] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization. Int. Conf. Learn. Represent., 2015.
[24] Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: a survey, IEEE Trans. Knowl. Data Eng. 14 (2020), https://doi.org/10.1109/tkde.2020.2981333, 1–1.
[25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. (2020) 1–21, https://doi.org/10.1109/TNNLS.2020.2978386.
[26] J. Li, D. Cai, X. He, Learning graph-level representation for drug discovery, Arxiv (2017) 1709.03741, http://arxiv.org/abs/1709.03741.
[27] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Comput. Aided Mol. Des. 30 (2016) 595–608, https://doi.org/10.1007/s10822-016-9938-8.
[28] K.T. Schütt, F. Arbabzadah, S. Chmiela, K.R. Müller, A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks, Nat. Commun. 8 (2017) 6–13, https://doi.org/10.1038/ncomms13890.
[29] K.T. Schütt, P.J. Kindermans, H.E. Sauceda, S. Chmiela, A. Tkatchenko, K. R. Müller, SchNet: a continuous-filter convolutional neural network for modeling quantum interactions, in: Adv. Neural Inf. Process. Syst., 2017, pp. 992–1002.

[30] O.T. Unke, M. Meuwly, PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges, J. Chem. Theory Comput. 15 (2019) 3678–3693, https://doi.org/10.1021/acs.jctc.9b00181.
[31] C. Chen, W. Ye, Y. Zuo, C. Zheng, S.P. Ong, Graph networks as a universal machine learning framework for molecules and crystals, Chem. Mater. 31 (2019) 3564–3572, https://doi.org/10.1021/acs.chemmater.9b01294.s001.
[32] K. Liu, X. Sun, L. Jia, J. Ma, H. Xing, J. Wu, H. Gao, Y. Sun, F. Boulnois, J. Fan, Chemi-Net: a molecular graph convolutional network for accurate drug property prediction, Int. J. Mol. Sci. 20 (2019) 3389, https://doi.org/10.3390/ijms20143389.
[33] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, M. Zheng, Pushing the boundaries of molecular representation for drug discovery with graph attention mechanism, J. Med. Chem. (2019), https://doi.org/10.1021/acs.jmedchem.9b00959 acs.jmedchem.9b00959.
[34] J. Klicpera, J. Groß, S. Günnemann, Directional message passing for molecular graphs, Arxiv (2020) 2003.03123, http://arxiv.org/abs/2003.03123.
[35] G. Li, M. Müller, A. Thabet, B. Ghanem, DeepGCNs: can GCNs go as deep as CNNs?, in: Int. Conf. Comput. Vis., 2019, pp. 9267–9276, http://arxiv.org/abs/1904.03751.
[36] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, R. Barzilay, Analyzing learned molecular representations for property prediction, J. Chem. Inf. Model. 59 (2019) 3370–3388, https://doi.org/10.1021/acs.jcim.9b00237.s001.
[37] J.M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N.M. Donghia, C. R. MacNair, S. French, L.A. Carfrae, Z. Bloom-Ackerman, V.M. Tran, A. Chiappino-Pepe, A.H. Badran, I.W. Andrews, E.J. Chory, G.M. Church, E.D. Brown, T. S. Jaakkola, R. Barzilay, J.J. Collins, A deep learning approach to antibiotic discovery, Cell 180 (2020) 688–702.e13, https://doi.org/10.1016/j.cell.2020.01.021.
[38] R. Ramakrishnan, P.O. Dral, M. Rupp, O.A. von Lilienfeld, Quantum chemistry structures and properties of 134 kilo molecules, Sci. Data 1 (2014) 140022, https://doi.org/10.1038/sdata.2014.22.
[39] G. Chen, P. Chen, C.-Y. Hsieh, C.-K. Lee, B. Liao, R. Liao, W. Liu, J. Qiu, Q. Sun, J. Tang, R. Zemel, S. Zhang, Alchemy: a quantum chemistry dataset for benchmarking AI models, Arxiv (2019) 1906.09427, http://arxiv.org/abs/1906.09427.
[40] Z. Wu, B. Ramsundar, E.N. Feinberg, J. Gomes, C. Geniesse, A.S. Pappu, K. Leswing, V. Pande, MoleculeNet: a benchmark for molecular machine learning, Chem. Sci. 9 (2018) 513–530, https://doi.org/10.1039/c7sc02664a.
[41] L. Ruddigkeit, R. Van Deursen, L.C. Blum, J.L. Reymond, Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, J. Chem. Inf. Model. 52 (2012) 2864–2875, https://doi.org/10.1021/ci300415d.
[42] Q. Sun, T.C. Berkelbach, N.S. Blunt, G.H. Booth, S. Guo, Z. Li, J. Liu, J.D. McClain, E.R. Sayfutyarova, S. Sharma, S. Wouters, G.K.L. Chan, PySCF: the Python-based simulations of chemistry framework, Wiley Interdiscip. Rev. Comput. Mol. Sci. 8 (2018), e1340, https://doi.org/10.1002/wcms.1340.
[43] J.E. Fey, Matthias and Lenssen, fast graph representation learning with PyTorch geometric. Int. Conf. Learn. Represent., 2019.
[44] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828, https://doi.org/10.1109/TPAMI.2013.50.
[45] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Comput. Vis. Pattern Recognit., 2018, pp. 7132–7141, https://doi.org/10.1109/CVPR.2018.00745.
[46] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Conf. Empir. Methods Nat. Lang. Process., 2014, pp. 1724–1734, https://doi.org/10.3115/v1/d14-1179.
[47] O. Vinyals, S. Bengio, M. Kudlur, Order matters: sequence to sequence for sets. Int. Conf. Learn. Represent., 2016.