Flexible Dual-Branched Message-Passing Neural Network for a Molecular Property Prediction

Jeonghee Jo, Bumju Kwak, Byunghan Lee,* and Sungroh Yoon*

ABSTRACT: A molecule is a complex of heterogeneous components, and the spatial arrangements of these components determine the whole molecular properties and characteristics. With the advent of deep learning in computational chemistry, several studies have focused on how to predict molecular properties based on molecular configurations. MA message-passing neural network provides an effective framework for capturing molecular geometric features with the perspective of a molecule as a graph. However, most of these studies assumed that all heterogeneous molecular features, such as atomic charge, bond length, or other geometric features, always contribute equivalently to the target prediction, regardless of the task type. In this study, we propose a dual-branched neural network for molecular property prediction based on both the message-passing framework and standard multilayer perceptron neural networks. Our model learns heterogeneous molecular features with different scales, which are trained flexibly according to each prediction target. In addition, we introduce a discrete branch to learn single-atom features without local aggregation, apart from message-passing steps. We verify that this novel structure can improve the model performance. The proposed model outperforms other recent models with sparser representations. Our experimental results indicate that, in the chemical property prediction tasks, the diverse chemical nature of targets should be carefully considered for both model performance and generalizability. Finally, we provide the intuitive analysis between the experimental results and the chemical meaning of the target.

INTRODUCTION

To design de novo compounds, all possible sets should be explored throughout a combinatorial space called chemical compound space (CCS). Quantum mechanics (QM) functions as a guide that narrows down this search space based on the first principles of molecular dynamics. Density functional theory (DFT) is the standard method for analyzing a molecular electronic structure from its numerical wave functions in QM to predict molecular behaviors.

In the past decade, machine learning (ML)-based methods have been developed and have played a key role in various QM tasks. For example, support vector machine (SVM), Gaussian processes, kernel ridge regression models, and other ML-based studies have been developed and applied to predict molecular properties or model molecular dynamics. Recently, with the advent of large-scale quantum-mechanical databases, deep learning (DL)-based models have become crucial in quantum machine learning (QML) studies. Deep neural network (DNN)-based methods have made it possible to model hidden complex relationships between molecular structures and their properties.

Early DNN-based QM studies explored various tasks such as DFT prediction, a simple molecule dynamics simulation, atomization energy prediction, or learning electronic properties. These models considered a single atom as a neuron of an input layer, with the frame of a one-body representation of a given system. Given a molecular geometry, a single atom can be represented with its atomic number and position, and multiple atoms can also be represented in the same way with their interactions, using their mutual distances or angles. However, standard MLP-based models cannot handle various sizes of chemicals. This limitation triggers a harmful effect on the model generalizability.

Graph neural networks (GNNs) can address these limitations by describing a molecule as a graph. GNNs can train both node and edge representations, which are naturally equivalent to atoms and atom–atom relationships present in a molecule. This graph-based molecular representation has...
become widely adopted in solving various molecule learning tasks. Several GNN-based models\textsuperscript{12–16} adopted a localized convolution kernel to learn each component of a molecule. These models are called graph convolutional networks (GCNs). The spatial embedding-based GCN model\textsuperscript{1} trains each local attribute iteratively and then aggregates these features to represent molecular properties.

The message-passing neural network (MPNN) framework\textsuperscript{19} has taken these approaches one step further. On the basis of a graph convolution, MPNNs learn node features in two types of phases: message-passing and update phases. In the message-passing phase, the hidden representations of nodes, neighboring nodes, and their edges are aggregated to represent the locality of a given node, called a message. Subsequently, the hidden features of these nodes are updated with the message and other features in the following update phase. The final step in a network is called a readout phase; all of the features are combined to yield the predictions. In the MPNN framework, both the message aggregation and the readout phase use a sum operator to gather local attributes and all atomic attributes, respectively. All MPNN models can be considered as a type of spatial GCNs because in these models each atom feature is updated with its localized convolutional filters.

Since the late 2010s, several message-passing-based DNN models have been developed, such as SchNet\textsuperscript{12,20} GDML\textsuperscript{13} PhysNet,\textsuperscript{15} MegNet,\textsuperscript{21} and DimeNet.\textsuperscript{22} SchNet and the subsequent model architectures suggested a block structure, which consists of a set of specific message-passing functions. They also adopted a continuous-filter convolution to learn intramolecular interactions that take the form of a continuous distribution. These models used atom types and atom–atom distances as inputs of the model. Some studies used more sophisticated functions to represent angles in a rotational equivariant approach.\textsuperscript{23,24} Many of the MPNN-based models exhibited competitive performances in various QM research areas, even in the molecular property prediction on a nonequilibrium state.

However, most of the previous works did not consider the natural heterogeneities among chemical properties. Although subtle, the target properties have originated from different chemical natures. For example, among the 12 targets from QM9, nine describe the types of given molecular energy, while the other three do not. A conventional DL optimization can select the better parameters for describing all of the targets. However, it cannot determine that feature type that is more important for predicting each target. In this case, the optimal parameters in the given model architecture may vary according to target types. Almost all previous models have no choice but to optimize parameters under the assumption that all types of features such as node features or edge features contribute equally to predicting all targets. This may limit the model generalizability and transferability because it ignores the diversity of chemical factors determining various properties.

In addition, GNNs have suffered from differentiating nodes, especially with deep architecture, that is, the oversmoothing problem.\textsuperscript{17,25} This problem emerges because iterative aggregations of neighboring features are equivalent to a repeated mixing of local features, eventually making all node representations too similar and reducing the model performance. Therefore, stacking more layers to a network cannot be the solution in the GNN (MPNN) case without careful consideration.

To overcome all these limitations, we propose a novel GNN for molecular property prediction, namely, a dual-branched Legendre message-passing neural network (DL-MPNN) with simple and powerful scalability. We used two types of features: an atom type and a position in three-dimensional space. We calculated the distance between two atoms and an angle between three atoms from atomic coordinates. In detail, we applied modified Legendre rational functions for radial basis functions of atom–atom distances. For a representation of angles between atoms, orthogonal Legendre polynomials were used. Finally, we adopted trainable scalars to each type of feature for model flexibility in multitask QM learning. Our approaches exhibited superior performances to other recent models in quantum-mechanical benchmark tasks. In addition, we validate the effectiveness of the proposed method via ablation studies.

To the best of our knowledge, this study is the first attempt to automatically explore a relative feature of importance for each target type without increasing the computational cost. The introduction of the trainable scalars has several advantages over that of using an additional layer. First, the overall computational cost is not significantly increased, because it only adopts scalar multiplications. Subsequently, it makes the model flexible for heterogeneous types of features in multitarget tasks.

In addition, we introduce an MLP-based novel-type block called a single-body block. These blocks are stacked to form a discrete branch in the model and do not communicate with the message-passing branch, as they solely train only atom embeddings. The outcome of this MLP-based pathway is aggregated with that of the MPNN at the final stage. Accordingly, we obtain an atom feature that is not overmixed by incoming messages during the training.

In summary, our contributions are as follows: (1) We propose a novel dual-branched network for molecular property prediction, which comprises a message-passing for locally aggregated features and a fully connected pathway for discrete atoms. (2) We introduce trainable scalar-valued parameters to enable the model to enhance more important feature signals according to each label. (3) Our experimental results are better than those of most previous works in the public benchmarks with sparser representations. In addition, we verify that various quantum-chemical factors contribute differently according to target molecular properties.

### Experimental Section

Our neural network is dual-branched, comprising two different types of networks. One is the MPNN, a type of spatial-based GCNs, and the other is the MLP-based network. The details are described in the following sections.

**Preliminaries.** We briefly introduce the notations and the data used in this study. We describe a molecule as a set of different types of i points $V = \{Z, P\}$ and $v_i = \{z_i, p_i\}, Z = \{z_i\}$ is a set of scalar-valued charges for each atom type. $P = \{p_i\}$, where $p_i = (p_i^x, p_i^y, p_i^z)$ is a set of $xyz$ coordinates of atom points in the Euclidean space. We denote $v_{ij} \epsilon \mathbb{N}(i)$ as a set of neighboring atoms of $v_i$.

**Data Preprocessing.** We used two types of information between three neighboring atoms; atom types $z_i$ and coordinates $p_i$ of each atom in the molecules. We calculated distances $d_{ij}$ and angles $\alpha_{ijk}$ between atom coordinates $p_i$, $p_j$, and $p_k$. Because coordinates are used to describe the whole molecular structure, a single position of an atom is mean-
ingless. We initialized random trainable atom type embeddings $X$ from $Z$ and trained them. All $d_{ij}$ and $\alpha_{ijk}$ values from the molecules were calculated before the model training. More details are described in Figure 1.

| Atom type | Atomic coordinate (xyz position) |
|-----------|----------------------------------|
| C         | [$p_{1x}, p_{1y}, p_{1z}$]       |
| O         | [$p_{2x}, p_{2y}, p_{2z}$]       |
| ...       | ...                              |

Figure 1. Data preprocessing. An atom type $Z$ and atomic coordinate $p$ were used in the model. We created trainable embeddings $X$ from $Z$ and calculated distances $d_{ij}$ and angles $\alpha_{ijk}$ from the coordinates $p_i, p_j, p_k$.

**Input Embedding: Distance and Angle Representations.**

**Distance Representation: Radial Basis Function.** We used radial basis functions to expand a scalar-valued edge distance to a vector. In particular, we adopted sequences of Legendre rational polynomials, one of the continuous and bounded orthogonal polynomials. The Legendre rational function $R_n(x)$ and Legendre rational polynomial $P_n(x)$ are defined as

$$R_n(x) = \sqrt{\frac{2}{x+1}}P_n\left(\frac{x-1}{x+1}\right)$$

(1)

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2-1)^n$$

(2)

where $n$ denotes the degree. The $n$th-order Legendre rational polynomials are generated recursively as follows.

$$R_{n+1}(x) = \frac{2n+1}{n+1}xR_n(x) - \frac{n}{n+1}R_{n-1}(x)$$

for $n \geq 1$ (3)

We selected the first $n$th order polynomials $R_1(x), R_2(x), ...$, $R_n(x)$, such that a scalar-valued distance $d \in \mathbb{R}$ is embedded as an $n$-dimensional vector $d \in \mathbb{R}^n$. The plot of functions degree of $1, 2, ..., 12$ are described in the top right side of Figure 2. We modified the equations to set the maximum value as 1.0, to make the distribution of the bases similar to that of the initial atom embedding distribution.

Recall that we did not use any molecular bond information. Following previous studies, we set a single scalar cutoff $c$ and assumed that any atom pair $v_i$ and $v_j$ located closer than the cutoff can interact with each other, and vice versa, before training. In other words, we build edge representations $e_{ij} = (z_i, z_j, d_{ij})$ between any two atoms close to each other. The cutoff value is analogous to the kernel height or width in the conventional convolutional layer because it determines the size of the receptive field of the local feature map.

**Angle Representation: Cosine Basis Function.** Similarly, we adopted another sequence of orthogonal polynomials to represent a scalar-valued angle $\alpha_{ijk}$ as vectors: Legendre polynomials of the first kind. Legendre polynomials of the first kind are the orthogonal polynomials that exhibit several useful properties. They are the solutions to the Legendre differential equation. They are commonly used to describe various physical systems and dynamics. Their formula is expressed as

$$Q_n(x) = \sum_{k=0}^{n} x^k \frac{n!}{k!(n-k)!} \frac{(n+k-1)!}{2^n}$$

(4)

The polynomials of degree $n$ are orthogonal each other, such that

Figure 2. Basis functions for data preprocessing. For any pair of two atoms located closer than a given cutoff, we created an edge representation between two atoms regardless of molecular bond information. A scalar-valued distance is expanded as an $n$-dimensional vector by radial basis functions. For radial basis functions, we used Legendre rational polynomials. Angle representations are depicted by any two edges sharing one atom. A cosine-valued angle is represented as the degree = 1, 2, ..., $n$th Legendre polynomials (top right).
\[
\int_{-1}^{1} Q_m(x)Q_n(x) \, dx = 0 \quad \text{if } n \neq m
\] (5)

We calculated angles \(\alpha_{ij}\) between any pair of two edges sharing one node, \(\vec{e}_{ij}\) and \(\vec{e}_{ik}\). We selected the first \(m\)th order polynomials \(Q_1(x), Q_2(x), \ldots, Q_m(x)\), such that a scalar-valued angle \(\alpha_{ijk}\) is embedded as an \(m\)-dimensional vector \(\alpha \in \mathbb{R}^m\). The scheme is presented on the left side of Figure 2. We calculated cosine values of each angle and embed them with Legendre polynomials.

Comparison of Two Different Types of Legendre Polynomials. We briefly compare the two different Legendre polynomials. First, the sequences of Legendre rational functions \(\{P_i(x)\}_{i=1,\ldots,n}\) are adopted as the encoding method for distances, which have different distributions over the distances. The standard deviations of the distributions of the distance encodings are higher at shorter distance values. Therefore, they can approximate the interatomic potentials, whereas distance encodings are higher at shorter distance values.

In the single body block \(S_t\), the output of \(I_t\) is expanded to a vector \(d_{ij}\) in the radial basis function \(\{(P_i(x))_{i=1,\ldots,n}\}\). The scalar \(d_{ij}\) is calculated from the edges. Then \(\alpha_{ijk}\) is encoded to an \(m\)-dimensional vector before training via Legendre polynomials of the first kind \(\{Q_k(x)\}_{k=1,\ldots,m}\). In this study, we set \(m = 12\).

In MPNN, the message passing and readout steps of our model can be summarized as

\[
m_{it}^{t+1} = f^m(t^{t+1}, h^{t}_{ij}, \alpha_{ijk}) \quad (t = 1, ..., T) \quad \text{(message passing)}
\]

\[
h^{t+1} = f^{t}_n(h^{t}_{ij}, m_{it}^{t+1}) \quad \text{(update)}
\]

\[
\hat{y} = f_{\eta}(\mathbb{V}h^{T}) \quad \text{(readout)}
\]

where \(m, h, e, \) and \(\alpha\) denote the message of atom \(v_i\), and its neighbors, single-atom representation of atom \(v_j\), interaction between two neighboring atoms \((v_i,v_j)\), and angle information between three atoms \((v_i,v_j,v_k)\), respectively. \(t\) is the time step or layer order in the model, and \(\hat{y}\) represents the predicted target value. Both \(f^m_{i}\) and \(f^t_{n}\) represent the graph convolution, and \(f_{\eta}\) is the sum operation.

In the proposed model, there are four types of blocks: the embedding block, output block, single body block, and interaction block. All blocks except the embedding block are sequentially stacked multiple times. Detailed explanations are presented in the next section.

Model Architecture. The overall model architecture is described in Figure 3. There are two discrete pathways in the network, which include the MLP-based pathway with single body blocks and MPNN-based pathway comprising output blocks and interaction blocks. First, an atom type \(Z_{ij}\) is embedded by the embedding block. The distances are calculated from atomic coordinates \(p_i\), and if a distance \(d_{ij}\) is less than the predefined cutoff value, then \(d_{ij}\) is represented as an edge attribute of the molecule graph. The scalar \(d_{ij}\) is expanded to a vector \(d_{ij}\) in the radial basis function \(\{(P_i(x))_{i=1,\ldots,n}\}\). The scalar \(\alpha_{ijk}\) is calculated from the edges. Then \(\alpha_{ijk}\) is encoded to an \(m\)-dimensional vector before training via Legendre polynomials of the first kind \(\{Q_k(x)\}_{k=1,\ldots,m}\). In this study, we set \(m = 12\).

In MPNN, the message passing and readout steps of our model can be summarized as

\[
m_{it}^{t+1} = f^m(t^{t+1}, h^{t}_{ij}, \alpha_{ijk}) \quad (t = 1, ..., T) \quad \text{(message passing)}
\]

\[
h^{t+1} = f^{t}_n(h^{t}_{ij}, m_{it}^{t+1}) \quad \text{(update)}
\]

\[
\hat{y} = f_{\eta}(\mathbb{V}h^{T}) \quad \text{(readout)}
\]

where \(m, h, e, \) and \(\alpha\) denote the message of atom \(v_i\), and its neighbors, single-atom representation of atom \(v_j\), interaction between two neighboring atoms \((v_i,v_j)\), and angle information between three atoms \((v_i,v_j,v_k)\), respectively. \(t\) is the time step or layer order in the model, and \(\hat{y}\) represents the predicted target value. Both \(f^m_{i}\) and \(f^t_{n}\) represent the graph convolution, and \(f_{\eta}\) is the sum operation.

In the proposed model, there are four types of blocks: the embedding block, output block, single body block, and interaction block. All blocks except the embedding block are sequentially stacked multiple times. Detailed explanations are presented in the next section.

Block Details. We adopted four types of blocks in the model. The embedding block \(E\), the output block \(O_t\), the single body block \(S_t\), and the interaction block \(I_t\) are introduced in the next section. Each block is introduced in the next section.

Embedding Block. In the embedding block \(E\), the categorical atom types \(Z_{ij}\) are represented as float-valued trainable features. \(Z_{ij}\) is embedded into a vector \(\vec{e}_{ij}\) that represents the atom types. These blocks handle the two-body representation.

Single Body Block. Message-passing functions would mix all representations of a molecule. We assumed that it is not always beneficial to predict molecular properties. Therefore, we introduced a separated path from a message-passing pathway. In the single body block \(S_t\), the output \(X_{ij}\) from \(E\) enters and is expanded to a vector \(d_{ij}\) in the radial basis function \(\{(P_i(x))_{i=1,\ldots,n}\}\). The scalar \(d_{ij}\) is calculated from the edges. Then \(\alpha_{ijk}\) is encoded to an \(m\)-dimensional vector before training via Legendre polynomials of the first kind \(\{Q_k(x)\}_{k=1,\ldots,m}\). In this study, we set \(m = 12\).

Output Block. In the output block \(O_t\), the output of \(I_t\) is trained with \(R_{ij}\) except the first block, which takes an embedded input \(X_{ij}\) from \(E\). \(\oplus\) and \(\otimes\) denote the direct sum and direct product, respectively. All blue-boxed objects are \(\gamma\)-trainable. These blocks train the two-body representation.

Interaction Block. In the interaction block \(I_t\), each output from the \(O_t\) and \(R_{ij}\) is applied. The blue boxes and the
operators are also used in these blocks. In addition, these blocks train the three-body interaction. 

Trainable Scalars: $\gamma$. In molecular property prediction tasks, various factors determine multiple properties with different contribution weights for each target. To tackle this issue, we did not consider any additional layer or attention preventing the model from the oversmoothing problem and becoming cost-intensive. Instead, we proposed a more simple solution. As a solution, we introduced trainable scalars $\gamma$ to each layer. This makes the contributions of heterogeneous factors flexible, according to each target. In other words, the importance of heterogeneous features such as atom type and atom–atom distances can be various over targets. Therefore, our model can focus on more important features of the target. Mathematical formulas are described below.

$$h^{t+1} = \sigma(Wh^t + b) \quad \text{(conventional dense layer)}$$

$$h^{t+1} = \gamma \sigma(Wh^t + b) \quad \text{(with trainable scalars)}$$

We introduced $\gamma$ to some of the layers in the output blocks, single body blocks, and the interaction blocks. All $\gamma$ values are trained independently of each other (subscripts were omitted for simplicity). We initialized $\gamma$ values from the exponential function of random normal distribution $\mathcal{N}(0.0, 0.1^2)$ ($\mu = 0.0$, $\sigma = 0.1$), namely, $\gamma \sim \mathcal{N}(0,0.0,0.1^2)$. This was because we observed that the slightly right-skewed distribution performed better than the normal distribution with $\mu = 1.0$, without skewness. This distribution is presented in Figure 5.

Related Works. Starting from DTNN, several GNNs adopt the perspective of a molecule as a graph. Under this perspective, a molecular system is a combination of atoms and many-body interatomic interactions that correspond to graph nodes and edges. MPNNs introduced the message concept, which is an aggregation of attaching edges and corresponding neighbor nodes. SchNet is a multiblock network based on the message-passing framework. The gradients flow atomwise, and the convolution process of atom–atom interaction features is in the interaction block. Subsequently, PhysNet,
MegNet, and MGCGN extended the previous works and improved the performances based on the message-passing multiblock framework.

More recent works introduced angle information to describe the geometry of a molecule. With angle information, we can inspect up to three-body interactions. Spherical harmonics were used to represent an original angle to be rotationally equivariant. Several studies introduced a Clebsch-Gordan decomposition to represent an angle comprising a linear combination of irreducible representations. Theoretically, it can be expanded to be arbitrary n-body networks, and most of the networks do not explore more than three-body interactions in limited computational resources. In summary, most of the recent GNNs on molecular learning used an atom type (one-body), a distance between an atom pair (two-body), and an angle between three atoms (three-body). As aforementioned, we also used the same input features in accordance with the previous GNNs, for a fair evaluation.

Training and Evaluation. Data Set. We used QM9 and molecular dynamics (MD) simulation data sets for the experiment. QM9 is the most popular quantum-mechanics database created in 2014, which comprises 134,000 small molecules made up of five atoms, which include carbon, oxygen, hydrogen, nitrogen, and fluorine. Each molecule has 12 numerical properties, such that the benchmark consists of 12 regression tasks.

The MD simulation data set is created for the energy prediction task from molecular conformational geometries. The simulation data are given as trajectories. The subtasks are divided according to each molecule type. The energy values are given as a scalar (kcal/mol), and forces are given as three-dimensional vectors (in xyz format) of each atom. The energies can be predicted solely on the molecular geometry or by using the additional forces. We used the most recent subdata sets of which the properties were created from a CCSD(T) calculation, which is a more reliable method than the conventional DFT method.

Implementation. The tensorflow package of version 2.2 was used to implement the proposed framework. To calculate Legendre polynomials, we utilized the scipy package ver. 1.5.2.

For the QM9 data set, we trained the model at least 300 epochs for each target. We terminated the training when the validation mean absolute error (MAE) did not decrease for 30 epochs. Therefore, the overall training epochs are slightly different from each label. We randomly split the train, valid, and test data set to 80%, 10%, and 10% of whole data set, respectively, in accordance with the guideline. The initial learning rate and decay rate were set to 10−3 and 0.96 per 20 epochs, respectively. Adam was used as the optimizer, and the batch size was set to 24.

For the MD simulation data set, most of the training configurations were the same with those of the QM9 experiment. However, we modified the loss function because the energies were trained using both energies and forces, which differs from QM9. We obtained the predicted forces from the gradients of the atom positions following from the previous works. We also followed the original guideline for data splitting. Therefore, we used 1000 samples to train all subtasks, the other 500 for a test except ethanol, and 1000 for a test of ethanol. We set the decay rate as 0.9, which makes the learning rate decrease faster than the QM9 training.

Evaluation. First, we evaluated the model performance with MAE, the standard metric of QM9. We compared the performance of our model with the models of previous studies SchNet, Cormorant, LieConv, DimeNet, and DimeNet++. We also analyzed our proposed methods in terms of the effect of γ values as well as the single body block of each target. We also analyzed the effect of the single body block. We evaluated the performance of the model without single body blocks (not dual-branched architecture).

In particular, we observed that the single body block contributed to the model performance differently with each target type. We inferred that these differences can be explained by the different nature of the properties. We discussed the relationship between some of the results and the chemical interpretations in the next section.

Table 1. Mean Absolute Error on QM9 Compared with Previous Works

| target | unit | SchNet | Cormorant | LieConv | DimeNet | DimeNet++ | DL-MPNN | MP only |
|--------|------|--------|-----------|---------|---------|-----------|---------|---------|
| μ      | D    | 0.033  | 0.038     | 0.032   | 0.0286  | 0.0297    | 0.0256  | 0.0238  |
| α      | bohr³ | 0.335  | 0.085     | 0.084   | 0.0469  | 0.0435    | 0.0444  | 0.0457  |
| ε_{HOMO} | eV   | 0.041  | 0.034     | 0.030   | 0.0278  | 0.0246    | 0.0223  | 0.0238  |
| ε_{LUMO} | eV   | 0.034  | 0.038     | 0.025   | 0.0197  | 0.0195    | 0.0169  | 0.0163  |
| Δε      | eV   | 0.063  | 0.061     | 0.049   | 0.0348  | 0.0326    | 0.0391  | 0.0403  |
| (R²)    | bohr³ | 0.073  | 0.961     | 0.800   | 0.331   | 0.331     | 0.414   | 0.385   |
| zPE    | meV  | 1.7    | 2.027     | 2.280   | 1.29    | 1.2        | 1.2     | 1.2     |
| U₀     | eV   | 0.014  | 0.022     | 0.019   | 0.00802 | 0.0063    | 0.0074  | 0.0084  |
| U      | eV   | 0.019  | 0.021     | 0.019   | 0.00789 | 0.0063    | 0.0074  | 0.0085  |
| H      | eV   | 0.014  | 0.021     | 0.024   | 0.00811 | 0.0065    | 0.0076  | 0.0092  |
| G      | eV   | 0.014  | 0.020     | 0.022   | 0.00898 | 0.0076    | 0.0076  | 0.0083  |
| Cₓ     | calmolK | 0.033  | 0.026     | 0.038   | 0.0249  | 0.0249    | 0.0234  | 0.0235  |

*DL-MPNN and MP only only denote our dual-branched model and the model without MLP-pathway, respectively.*
We compared the change in ratio of the average of all γ values from the single block and that of the output blocks over epochs throughout the training. We extracted the γ values and calculated the ratio of $\frac{\sum (\gamma \text{ values from the single block})}{\sum (\gamma \text{ values from the output block})}$ every 30 epoch, until 200 epochs. After this point, the changes in the ratios varied negligibly, so we did not depict the ratios after that time point. Note that we did not compare the γ values directly among different layers or targets. The magnitudes of these values are determined by several complex sources: the original feature scale, layer weights, layer biases, activation functions, and uncontrollable random noises. Therefore, the γ values themselves cannot be evaluated directly.

## RESULTS AND DISCUSSION

### Model Performance

#### The Comparative Results

We compared the performance of our model with that of other MPNN-based models that also used atom types and locations as inputs. We analyzed six of 12 targets in QM9. We achieved advanced performances in six of the 12 targets. For the MD simulation, our model exhibited the best performance among other models in four of the 12 targets. For QM9, our model achieved advanced performances in six of the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). Gibbs free energy ($G$), and others. These properties are relevant in molecular reactions to external effects. However, predictions for other targets such as the electronic spatial extent ($⟨R^2⟩$) and internal energies ($U_{0}$, $U$, $H$) were not superior to those of other models.

We found that this may be related to the distribution of each target value. We observed that, when the mean of the target value is closer to zero and the standard deviation of the target value is smaller, the prediction performances are better. For example, the 95% confidence intervals (CIs) of the distribution of the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) are $−6.5 \pm 1.2$ and $0.3 \pm 1.3$, respectively. In case of internal energies including $U$, $U_{0}$, and $H$, the 95% of CI is $−76.5 \pm 10.4$, $−76.1 \pm 10.3$, and $−77.0 \pm 10.5$, respectively. This issue triggered by the diverse target distribution of QM9 has already been reported in the literature.

#### The Effect of Basis Functions

We briefly discuss the effect of Legendre polynomials as radial basis functions in our model. The results are shown in Table 3. We compared our radial basis functions with Bessel basis functions in DimeNet and DimeNet++ using QM9. We analyzed six of 12 targets in QM9, $μ$, HOMO, LUMO, gap, ($⟨R^2⟩$), and $U$. To summarize the comparisons, only $μ$ showed clearly improved validation performances (~30% lower MAE), and the other five targets showed slightly decreased MAE values.

This inconsistency over targets could not be clearly explained. Therefore, we concluded that a Legendre polynomial of our model is not worse than Bessel basis functions. Note that our cutoff value is less than those of any other previous works, maintaining the model performances. This will be discussed in the next section.

### The Effect of Trainable Scalars

#### The Role of the $γ$ in the Model Performance

We trained our model with and without trainable γ values for all 12 labels in QM9 data sets. We found that the validation performances were significantly improved (over 10%) in the case of $μ$, LUMO, and ($⟨R^2⟩$) with γ values. Overall, the validation MAE values with γ values were lower than those without γ values in nine target cases in QM9. With $U_{0}$ and $U$, the MAEs without training γ values are 3% lower than those with training γ. The results are shown in Figure 6.

#### Adaptive Dependency on Pathways by Targets

We analyzed $\frac{\sum fi^2}{\sum o^2}$, which is the ratio of the sum of the trained coefficients from the MLP pathway (the last single body block) over those from the MPNN pathway (the last output block), respectively, by each target. We plotted the ratio changes with training steps in Figure 7. If the ratio increases during the training, predictive performances (GMAE) tend to improve. If the ratio decreases or remains constant during the training, predictive performances (GMAE) tend to degrade.
training steps, the MLP pathway is more critical than the MPNN pathway for target prediction, and vice versa. Note that the weights of the MLP pathway are only trained on atom types $Z$.

We found that, in training the model on $\mu$, the single body blocks were less important. During the training on the target $\mu$, the trainable scalars in $S_t$ become close to zero, while other targets were not the cases. We found that this exceptional result can be explained with the construction of $\mu$, which is the sum of dipole moments of all atom pairs in a molecule. We repeated the experiment with $\mu$ three times and observed these patterns in all the times (results are not shown).

The formulation of the dipole moment is given by

$$
\mu = \sum_i q_i R_i
$$

where $q$ and $R$ are charges and position vectors, respectively. A dipole moment occurs between two polarized atoms, especially in a covalent bond or an ionic bond. By definition, dipole moments are represented by the charge separation between atom pairs (dipoles) in a molecule. The atom-only representations did not play a key role in the case of predicting $\mu$ in our experiments, which is consistent with the formulation. The overall performances also support these observations (Table 1). In the case of $\mu$, the performance was better (MAE $= 0.0238$) when the model used the MPNN pathway only than when the full model was used (MAE $= 0.0256$). However, the full model showed better performance than the MPNN pathway only model, in most of the target predictions. It also confirms that the atom-only representations from the MLP pathway does not help in predicting the $\mu$ of given molecules.

In addition, the plots in Figure 7 in predicting $U_0$, $U$, and $H$ are similar to each other. $U_0$ and $U$ mean the internal energy at 0 and 298.15 K, respectively. $H$ is the enthalpy at 298.15 K, which has the internal energy terms. These targets are closely related to each other by construction. Considering the experimental results, we concluded that the model trained differently with respect to the target, and the pattern of trained results is consistent with the nature of the chemistry.

Density of a Molecule Representation. As mentioned before, the cutoff $c$ value determines the node connections in a molecular graph. With a high cutoff value, the molecular graph representation would be dense. When the cutoff is low, the corresponding graph representation would be sparse.

The dense graphs generally have higher feasibilities in capturing the connectivity information in a graph than the sparse graphs. However, dense structures are exposed to higher risks of gathering excessive features even when there is less necessary information. Besides, in an MPNN, the messages of every node from its neighboring edges are mixed repeatedly throughout the network. If a graph is excessively dense, most messages become indistinguishable, because all atoms would refer all other neighbors in every step. This may potentially increase the risk of an oversmoothing problem, prone to occur in deep GCNs.
It is challenging to determine the most appropriate cutoff value of the graph data set; however, we found a clue from molecular conformations. We observed that the average distance between any atom pair in QM9 molecules is 3.27 Å (Å). The atom pairs within 4.0 Å, which is the cutoff value used in this work, account for 72% of all the pairs in the QM9 data set. Previous models SchNet, MegNet, and DimeNet/DimeNet++ adopted 10.0, 5.0, and 5.0 as their cutoff values, respectively. We observed that 99.99% and 89% for cutoff values of 10.0 and 5.0 of the distances, respectively, are represented in molecular graphs.

Note that the angle representation is defined with two different edges sharing one atom. This indicates that, for the angle representation, the required number of computations increases linearly with the squared number of edges. If the number of nodes of a graph are \( |V| \), then the upper bound of the number of representations including all nodes, edges, and angles of the graph is given as \( |V|^2 \), even if the message directions were not considered. We compared graph density values according to different cutoff values. If the edges are within a cutoff value of 4.0, then the number of edge representations are reduced to half ((72%)² = 52.8%) of the overall possible number of edges \( |V|^2 \). If the cutoff value is 5.0, then the proportion of the number of the edge representations is increased to 80% ((89%)² = 79.2%).

We demonstrate that the cutoff value of 4.0 Å is sufficient in describing a molecular graph in QM9. As mentioned earlier, the molecules in QM9 comprise five atoms, namely, C, H, O, N, and F. The lengths of the covalent bonds between any two atoms among these five atoms are always less than 2.0 Å. Because the maximum length of any two successive covalent bondings is less than 4.0, the model can capture any two-hop neighboring atoms simultaneously, with a cutoff value of 4.0 Å. This property makes it possible to identify all the angles between two covalent bonds. The detailed description is presented in Figure 8. From these observations, we argue that the 4.0 Å is the best choice as the cutoff value for an efficient training and chemical nature. The same holds for MD simulation as well, because in this data set, all the atoms are carbon, hydrogen, and oxygen (C, H, and O). Finally, we argue that 4.0 Å is also an adequate value for organic molecules in the real world, because most molecular bond lengths are shorter than 2.0 Å.

**CONCLUSION**

In this study, we developed a novel dual-branched network, message-passing-based pathway for atom–atom interactions and fully connected layers for single atom representations. We represented a molecule as a graph in three-dimensional space with atom types and atom positions. We embedded a scalar-valued atom–atom distance as a vector using Legendre rational functions. Similarly, we embedded a scalar-valued angle as a vector using orthogonal Legendre polynomials. Both functions are complete orthogonal series and require low computational costs with recursive relations. Our model exhibited remarkable performances in two quantum-chemical data sets. In addition, we proposed trainable scalar values, such that the proposed model can attend more significant features according to the various natures of the targets during the training. With the analysis of the trained scalars, we also showed that the model can obtain an important interpretation ability of the target. We found that the trained scalar values can explain the chemical nature of the target. Furthermore, we adopted a smaller cutoff value than those used in previous MPNN models and showed that, with this value, we can save the computational resources without a loss of performance. Furthermore, we argued that this cutoff value is sufficiently long to identify the local structure of a molecule. Although we showed both the model performance and a hint of interpretability, our model was applied to restricted fields of small molecules. We will enhance the model scalability to broader applications including biomolecules, drugs, and crystals in our future works. Furthermore, we will conduct further analysis on the predictability of the model according to target distributions. Future works will be focused on other molecular property predictions or the predictions for more complex molecules in more various fields.

**AUTHOR INFORMATION**

**Corresponding Authors**

Byunghan Lee – Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Nowon-gu, Seoul 08826, Republic of Korea; orcid.org/0000-0002-6727-0975; Phone: +82-2-970-9019; Email: bhlee@seoultech.ac.kr

Sungroh Yoon – Department of Electrical and Computer Engineering, Seoul National University, Gwanak-gu, Seoul 08826, Republic of Korea; orcid.org/0000-0002-2367-197X; Phone: +82-2-880-1406; Email: sryoon@snu.ac.kr

**Authors**

Jeonghee Jo – Bio-MAX Institute, Seoul National University, Gwanak-gu, Seoul 08826, Republic of Korea; orcid.org/0000-0003-1154-1761

Bumju Kwak – Recommendation Team, Kakao Corporation, Seongnam-si, Gyunggi-do 13494, Republic of Korea; orcid.org/0000-0002-6517-1711

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.1c05877
Notes

We used two public data sets, namely, qm9 and MD simulation data sets. Specifically, we retrieved the qm9 data set from the open resource from DimeNet and DimeNet++. Our code for the model object is available at https://github.com/fromjade/dlmpn.git.

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2021 (2021M3A7B4911115), National Research Foundation of Korea, grant funded by the Korean Government (MSIT) (2018R1A2B3001628, 2019R1G1A1033253), K-BIO KIURI Center program (2020M3H1A1073304), and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. 2021-0-01343, Artificial Intelligence Graduate School Program in Seoul National University).

ADDITIONAL NOTE

Following from a GNN survey paper, we denote a spatial-based GCN as a type of GCN which uses a convolution filter on local context of a node. The MPNN is also a type of spatial-based GCN.

REFERENCES

(1) Gao, T.; Sun, S.-L.; Shi, L.-L.; Li, H.; Li, H.-Z.; Su, Z.-M.; Lu, Y.-H. An accurate density functional theory calculation for electronic excitation energies: The least-squares support vector machine. J. Chem. Phys. 2009, 130, 184104.
(2) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. Phys. Rev. Lett. 2010, 104, 136403.
(3) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. Phys. Rev. Lett. 2012, 108, 058301.
(4) Snyder, J. C.; Rupp, M.; Hansen, K.; Müller, K.-R.; Burke, K. Finding density functionals with machine learning. Phys. Rev. Lett. 2012, 108, 253002.
(5) Mills, M. J.; Popelier, P. L. Intramolecular polarisable multipolar electrostatics from the machine learning method Kriging. Comput. Theor. Chem. 2011, 975, 42–51.
(6) Cova, T. F.; Pais, A. A. Deep learning for deep chemistry: optimizing the prediction of chemical patterns. Front. Chem. 2019, 7, 809.
(7) Balabin, R. M.; Lomakina, E. I. Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies. J. Chem. Phys. 2009, 131, 074104.
(8) Houding, S.; Liem, S.; Popelier, P. A polarizable high-rank quantum topological electrostatic potential developed using neural networks: Molecular dynamics simulations on the hydrogen fluoride dimer. Int. J. Quantum Chem. 2007, 107, 2817–2827.
(9) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; von Lilienfeld, O. A.; Tkatchenko, A.; Müller, K.-R. Assessment and validation of machine learning methods for predicting molecular atomization energies. J. Chem. Theory Comput. 2013, 9, 3404–3419.
(10) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Machine learning of molecular electronic properties in chemical compound space. New J. Phys. 2013, 15, 095003.
(11) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. Phys. Rev. Lett. 2007, 98, 146401.
(12) Schütt, K. T.; Kindermans, P.-J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. Advances in Neural Information Processing Systems 2017, 992–1002.
(13) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. Sci. Adv. 2017, 3, e1603015.
(14) Konrod, R. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. arXiv preprint 1803.01388, 2018.
(15) Unke, O. T.; Mew Lynch, M. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. J. Chem. Theory Comput. 2019, 15, 3678–3693.
(16) Wang, X.; Li, Z.; Jiang, M.; Wang, S.; Zhang, S.; Wei, Z. Molecule property prediction based on spatial graph embedding. J. Chem. Inf. Model. 2019, 59, 3817–3828.
(17) Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. AI Open 2020, 1, 57–81.
(18) Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P. S. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 2021, 32, 4.
(19) Gilmer, J.; Schoenholz, S. S.; Riley, F. P.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. International Conference on Machine Learning 2017, 1263–1272.
(20) Schütt, K. T.; Sauceda, H. E.; Kindermers, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet-A deep learning architecture for molecules and materials. J. Chem. Phys. 2018, 148, 241722.
(21) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 2019, 31, 3564–3572.
(22) Klicpera, J.; Groß, J.; Günnemann, S. Directional message passing for molecular graphs. International Conference on Learning Representations 2020.
(23) Anderson, B.; Hy, T.-S.; Konrod, R. Cormorant: Covariant molecular neural networks. Advances in Neural Information Processing Systems; MIT Press, 2019.
(24) Müller, B. K.; Geiger, M.; Smidt, T. E.; Noé, F. Relevance of rotationally equivariant convolutions for predicting molecular properties. arXiv preprint 2008.08461 2020.
(25) Li, Q.; Han, Z.; Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. Proceedings of the AAAI Conference on Artificial Intelligence. AAAI2018; pp 3538–3545.
(26) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. Nat. Commun. 2017, 8, 1–8.
(27) Lu, C.; Liu, Q.; Wang, C.; Huang, Z.; Lin, P.; He, L. Molecular property prediction: A multilevel quantum interactions modeling perspective. Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33, 1052–1060.
(28) Poulenard, A.; Rakotosaona, M.-J.; Ponty, Y.; Ovsjanikov, M. Effective rotation-invariant point cnn with spherical harmonics kernels. 2019 International Conference on 3D Vision (3DV). 2019, 47–56.
(29) Smidt, T. E. Euclidean symmetry and equivariance in machine learning. Trends Chem. 2021, 3, 82.
(30) Konrod, R.; Lin, Z.; Trivedi, S. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. Advances in Neural Information Processing Systems; MIT Press, 2018.
(31) Fuchs, F. B.; Worrall, D. E.; Fischer, V.; Welling, M. SE (3)-transformers: 3D roto-translation equivariant attention networks. Advances in Neural Information Processing Systems; MIT Press, 2020.
(32) Klicpera, J.; Giri, S.; Margraf, J. T.; Günemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. NeurIPS-W; 2020.
(33) Oxtoby, D. W.; Gillis, H. P.; Butler, L. J. Principles of modern chemistry; Cengage Learning, 2015.
(34) Brau, C. A. Modern problems in classical electrodynamics; Oxford University Press, 2004.
(35) Daley, R. Organic Chemistry, Part 3 of 3; Lulu.com, 2005.
(36) Allen, F. H.; Kennard, O.; Watson, D. G.; Brammer, L.; Orpen, A. G.; Taylor, R. Tables of bond lengths determined by X-ray and neutron diffraction. Part I. Bond lengths in organic compounds. J. Chem. Soc., Perkin Transactions 2 1987, S1–S19.
(37) Anli, F.; Gungor, S. Some useful properties of Legendre polynomials and its applications to neutron transport equation in slab geometry. Appl. Math. Model. 2007, 31, 727–733.
(38) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. Sci. data 2014, 1, 1–7.
(39) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. Nat. Commun. 2018, 9, 1–10.
(40) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Lessing, K.; Pande, V. S. MoleculeNet: A Benchmark for Molecular Machine Learning. Chem. Sci. 2018, 9, 513.
(41) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. International Conference on Learning Representations, 2015.
(42) Finzi, M.; Stanton, S.; Izmailov, P.; Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. International Conference on Machine Learning, 2020; pp 3165–3176.
(43) Harris, C. R.; et al. Array programming with NumPy. Nature 2020, 585, 357–362.
(44) Jones, E.; Oliphant, T.; Peterson, P.; et al. SciPy: Open source scientific tools for Python. 2001; http://www.scipy.org/.
(45) Abadi, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; https://www.tensorflow.org/, Software available from tensorflow.org.