Neurocomputing 445 (2021) 12–25

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-view spectral graph convolution with consistent edge attention for molecular modeling

Chao Shang [a], Qinqing Liu [a], Qianqian Tong [a], Jiangwen Sun [b], Minghu Song [c,*], Jinbo Bi [a,c,*]

[a] Department of Computer Science and Engineering, University of Connecticut, CT, USA
[b] Department of Computer Science, Old Dominion University, VA, USA
[c] Department of Biomedical Engineering, University of Connecticut, CT, USA

ARTICLE INFO

Article history:
Received 6 April 2020
Revised 4 January 2021
Accepted 8 February 2021
Available online 2 March 2021
Communicated by Zidong Wang

Keywords:
Graph convolutional networks
Molecular graphs
Edge attention
Graph representation

ABSTRACT

Although graph convolutional networks (GCNs) that extend the convolution operation from images to graphs have led to competitive performance, the existing GCNs are still difficult to handle a variety of applications, especially cheminformatics problems. Recently multiple GCNs are applied to chemical compound structures which are represented by the hydrogen-depleted molecular graphs of different size. GCNs built for a binary adjacency matrix that reflects the connectivity among nodes in a graph do not account for the edge consistency in multiple molecular graphs, that is, chemical bonds (edges) in different molecular graphs can be similar due to the similar enthalpy and interatomic distance. In this paper, we propose a variant of GCN where a molecular graph is first decomposed into multiple views of the graph, each comprising a specific type of edges. In each view, an edge consistency constraint is enforced so that similar edges in different graphs can receive similar attention weights when passing information. Similarly to prior work, we prove that in each layer, our method corresponds to a spectral filter derived by the first order Chebyshev approximation of graph Laplacian. Extensive experiments demonstrate the substantial advantages of the proposed technique in quantitative structure-activity relationship prediction.

1. Introduction

Convolutional Neural Networks (CNNs) [1–3] have been powerful tools to model data of a grid-like structure, e.g., image, video, and speech. CNNs offer an efficient way to extract local stationary structures and features that are shared across the different areas of an object (e.g., an image) [4]. However, a broad range of scientific problems [5,6] generate data that naturally lie in irregular grids with non-Euclidean metrics, such as, those in computational chemistry, social studies and telecommunication networks, which are in the form of graphs or networks. The generalization of CNNs from images to graph inputs is not straightforward. Due to the lack of global parameterization, a common system of coordinates, vector space structure, or shift-invariance properties [7], the classical convolutions which use fixed filter size and stride distance cannot be applied directly to graphs that have arbitrary structures. It has been successful to create Graph Convolutional Networks (GCNs) [8,9] that can tackle tasks such as manifold analysis [10], predictions of user preference and connectivity in social network [11], and representations of network nodes [12–14].

In recent years, there has been a growing interest in incorporating deep learning approaches into chemoinformatics studies [8,15–18]. For example, the quantitative structure–activity relationship (QSAR) is a problem of predicting the activity, reactivity, or property of an unknown set of molecules based on analysis of an equation connecting the structures of molecules to their respective measured activity property [19]. Most QSAR studies have employed certain hand-crafted molecular descriptors or fingerprints [20]. In order to extract better representations for the chemical structure and bonding, a chemical compound can be expressed as a hydrogen-depleted molecular graph whose nodes correspond to (non-hydrogen) atoms while edges with discrete attributes represent chemical bonds. Deep graph convolutional networks have emerged recently to derive representations directly using molecular graphs [21–24]. However, current GCN methods have many limitations when handling the molecular graphs.

* Corresponding authors at: Department of Computer Science and Engineering, University of Connecticut, CT, USA (J. Bi); Department of Biomedical Engineering, University of Connecticut, CT, USA (M. Song).

E-mail addresses: chao.shang@uconn.edu (C. Shang), qinqing.liu@uconn.edu (Q. Liu), qianqian.tong@uconn.edu (Q. Tong), jsun@odu.edu (J. Sun), minghu.song@uconn.edu (M. Song), jinbo.bi@uconn.edu (J. Bi).

https://doi.org/10.1016/j.neucom.2021.02.025
0925-2312/

The key limitation is that existing approaches ignore the general consistency of the bond energies (enthalpies) and bond lengths (interatomic distances) in various molecules [25,26]. The bond dissociation energy, also known as bond enthalpy, is defined as the standard enthalpy change when a bond is cleaved by homolysis. Thus, the bond energy is the amount of energy required to break a bond. The equilibrium bond length is the average distance between the nuclei of two bonded atoms in a molecule. The higher the bond energy, the "stronger" the bond is between the two atoms, and the length of bond is smaller. The bond energies and bond lengths between two specific atoms are generally consistent cross different molecules. The tables about bond lengths (interatomic distances) [27] and bond energies [28] have been used in some standard handbooks, which is consistent for all molecules. For example, the bond length of the atom pair Carbon–Nitrogen (C–N) is 147 pm and the bond length of Carbon–Carbon (C–C) is 154 pm. In addition, the bond length is also affected by other factors, such as bond order. For example, the bond length of C = C is 134 pm and the bond length of C≡C is 120 pm. Overall, this means that the edges in different molecular graphs present some consistency, which we call "edge consistency". We aim to design a new GCN with a constraint to enforce that information passes along similar edges with similar attention weights.

In this work, we propose a consistent Edge-Aware multi-view spectral GCN (*EAGCN*) approach to enhance the molecular graph property prediction (also corresponding to QSAR prediction) by learning more accurate molecular representations. Our model enforces an edge consistency constraint that similar edges should have similar weights across all molecular graphs in a dataset. In addition, our model provides a new way to take advantage of the discrete edge attributes by creating multi-view graphs.

First, to design an edge consistency constraint, we explore the attributes of edges in molecular graphs. As illustrated in Table 1, these discrete attributes are important to describe an edge or bond [29]. Each discrete value of an edge attribute is considered as an edge type. For instance, an edge can be labeled in terms of whether the bond is formed between the Carbon and Oxygen atoms (i.e., the C-O bond) or whether the bond is aromatic. These attributes of edges are highly related with the bond lengths and energies. In the molecular graph, edges with the same attribute value have similar bond lengths and energies.

Based on these edge attributes, we present a consistent edge mapping (CEM) mechanism to learn the consistent attention weights of edges in order to enforce the edges with the same type to pass information using the same weight . The learned real-valued attention matrix assigns a non-negative weight to an edge, and assigns 0 when there is no edge between two nodes. Importantly, an effective way is provided to enforce consistent attention weights by building edge mapping dictionaries shown in Section 4.1. The same weight from the edge dictionary will be provided if two edges have the same edge attribute value. The weights in these edge dictionaries, which are the parameters in the GCN, are shared across different molecules, thus promoting *EAGCN* to learn the inherently invariant properties in the graphs.

Table 1
Edge attributes commonly used in molecular graphs.

Attribute	Description
`Atom Pair Type`	Defined by the type of the atoms that a bond connects (e.g., C–C, C-O).
`Bond Order`	Bond order (single bond, aromatic bond, double bond and triple bond).
`Aromaticity`	Is aromatic.
`Conjugation`	Is conjugated.
`Ring Status`	Is in a ring.

In graph theory, graph invariant is defined as a property preserved under any possible isomorphism of a graph, which includes the order invariant, permutation invariant and pair order invariant [24,30].

Second, because there are multiple attributes for each edge as shown in Table 1 [29], we model a molecular graph as multi-view graphs to utilize different edge attributes, where each attribute including a group of discrete values corresponds to a separate view. Hence, a molecular graph can be decomposed into multiple graphs from different views according to each specific edge attribute. The common GCN methods predefine a single fixed adjacency matrix A for a graph, which cannot catch the different edge attributes. In this paper, the discrete values of one attribute are used to create one edge mapping dictionary including attention weights for the corresponding view. Each view will employ its own edge (relation) consistent attention mechanism.

Furthermore, a new spectral filter is derived to rely on two coefficients of the first order Chebyshev approximation rather than just one as in the standard GCNs - the 0^{th} order and 1^{st} order coefficients (θ_0 and θ_1). These two coefficients are enforced to be opposed to each other ($\theta_0 = -\theta_1$) in all early spectral graph convolution [13]. We relax this requirement, resulting in more general and effective parameterization of the GCN network. To create a fingerprint for a molecule, besides learning the node representation (for each atom), learning a graph representation for the entire molecule is also important. We apply and compare two ways including simple sum and differential pooling [31] to integrate node embeddings into a graph representation. Finally, *EAGCN* provides an alternative fingerprint for chemical compounds other than hand-crafted ones, and prove to be useful in predicting the properties of the compounds in the experiments. Our code and data are available at https://github.com/Luckick/EAGCN.

Major contributions are summarized as follows:

1. We propose a consistent edge-aware multi-view spectral GCN model, namely, *EAGCN,* that preserves the edge (bond) consistency in molecular graphs. The proposed CEM mechanism in *EAGCN* learns the consistent edge attention weights cross molecules by building global edge dictionaries.
2. Multi-view graphs have been employed in our *EAGCN* model to take advantage of the discrete edge attributes. Here each edge attribute is characterized in a view of the molecular graph to explore the different attributes of edges in molecular graphs.
3. A new spectral filter derived from the Chebyshev approximation is designed for the spectral graph convolution. This spectral convolution operation allows more flexible and general network parameterization.

2. Related work

In a similar spirit to CNNs, GCN methods aggregate neighboring information based on the connectivity of the graph through filters. GCN methods [32] are classified into two categories according to the way of aggregating neighbor information: spatial methods [21,33,34] and spectral methods [35,36,13,37,38]. In the spatial way, the convolution at a graph node is carried out by summing up the embeddings of adjacent nodes according to weights provided by a filter. In spectral GCNs, node attributes are viewed as graph signals, and the convolution operation is defined through the Fourier transform which is derived from graph Laplacian and its eigen-space. The localized convolutional filters extracted local features independently of the spatial locations motivate the spatial graph convolutional architecture via first-order approximation of spectral graph convolutions.

2.1. Spectral graph convolutions

GCNs were first proposed in [35] where graph convolution operations were defined in the Fourier domain. Since the eigendecomposition of the graph Laplacian was needed, it required intense computation. To tackle this issue, smooth parametric spectral filters [39] were introduced to achieve localization in the spatial domain and the computational efficiency. In particular, Chebyshev polynomials [36] and Cayley polynomials [40] were utilized in these convolutional architectures to efficiently produce localized filters. Recently, Kipf et al. [13] simplified these spectral methods by a first-order approximation of the Chebyshev polynomials. Their derivation finally led to localization of one-step neighbors and achieved state-of-the-art performance.

However, the spectral filters learned by the above methods depend on the Laplacian eigenbasis which is linked to a fixed graph structure. Thus these models can only be trained on a single graph, and cannot be directly used to model a set of graphs with different structures.

Recently graph attention networks [41] were proposed to deal with arbitrary graphs when the graph structures were not fully known. The attention mechanisms allowed the model to deal with input graphs of varying size by assigning different weights to different nodes within a neighborhood while dealing with various sized neighborhoods. It determined the attention weight between any two nodes by calculating the similarity of the embeddings (or feature vectors) of the two nodes. It has not considered the case where edge attentions themselves are learnable. We design a new edge attention mechanism to understand which types of edges (bonds) are important for the target molecular property. Because even two nodes (atoms) are not similar, their edge (bond) can be important for a QSAR property, and should receive high attention weight without relying on node similarity as done in [41].

2.2. Non-spectral graph convolutions

The spatial graph convolution approaches [12,21,42] define convolutions directly on graph, which sum up node features over all spatially close neighbors by multiplying an adjacency matrix which is pre-defined rather than learnable. The varying sizes of the neighborhoods impose challenge on sharing the weights across graphs. Duvenaud [21] presented a CNN that operated directly on raw molecular graphs, and learned a specific weighted matrix for each node degree. Another approach [33] selected fixed-size neighbors and normalized these nodes to enable the traditional CNNs to be applied to graph inputs directly. In order to handle graphs of varying size and connectivity, Simonovsky et al. [43] proposed the edge convolutional network (ECC) for point cloud classification. Their model defined several node feature filters based on the edge label. Recently Hamilton et al. [12] introduced the task of inductive node classification, where the goal was to classify nodes that have not been seen during training. This approach sampled a fixed-size neighborhood for each node and achieved state-of-the-art performance on several datasets.

Recently, there has been an increasing interest in applying different attention mechanisms to GCN models [32,44]. The graph attention network (GAT) [41] first proposed attention mechanisms on graph convolutions to learn the relative weights between two connected nodes. The GAT approach determined the weight for an edge based on the similarity of the two nodes of the edge. Gated Attention Network (GAAN) [45] proposed a self-attention mechanism to compute an additional attention score for controlling each attention head's importance. Abu-El-Haija et al. [46] presented an attention mechanism for learning the context distribution based on the mathematical equivalence between the context distribution and the power series of the transition matrix. ASTGCN [47] further designed a spatial attention function and a temporal attention function to learn both spatial and temporal dependencies. AttentiveFP [48] was an extension of the GAT, and learned both local and nonlocal properties of a given chemical structure by combining GAT with the gated recurrent unit (GRU). The GROVER method [49] integrated message passing methods into the Transformer-style architecture to learn structural and semantic information of molecules from enormous unlabelled molecular data. However, all previous methods ignored the "edge consistency" illustrated in Section 3 when modeling the molecular graphs. We design a new consistent edge constrained attention mechanism to reserve the consistency of the bond energies and bond lengths on multiple molecules.

2.3. Convolutions on molecular graphs

Neural networks and GCNs have been applied to chemical studies such as protein interface prediction [16], and molecular representation and prediction [21,23,24,29,50]. A CNN was presented in [21] that operated directly on raw molecular graphs and generalized standard molecular feature extraction methods based on circular fingerprints (ECFP) [51]. Based on an autoencoder model, the method in [19] converted discrete representations (e.g., ECFP) of molecules to a multidimensional continuous one. Another work [24] attempted to use different attributes of chemical bonds, and graph edges were associated with attributes such as atom-pair type or bond order and modeled via tensor computation. The resultant graph network could utilize properties of both nodes (atoms) and edges (bonds). The method in [29] learned atomic embeddings which were then concatenated by the related bond features to form atom-bond feature vectors. In these works, node features and bond attributes were treated equally in the neighborhood aggregation. However, different types of bonds may play very different roles, and should receive different attention in relation to a target property.

3. Problem formulation

Let $G = (V, E)$ be an undirected graph where V is a set of n nodes, $E \subseteq V \times V$ is a set of n_E edges. The node features in a graph form a feature matrix $X \in \mathbb{R}^{n \times f_1}$ which is the input of the first layer so that we get $X^1 = X$. The edge features from a graph are represented by $\mathbb{Z} = \{Z_1, \ldots, Z_k\}$, where k is the number of edge features as shown in Table 1 and each $Z_i \in \mathbb{Z}$ has multiple discrete values. Each molecule is represented by such a graph G and is labeled by a molecular property $y \in Y$. The task of QSAR is to predict the target molecular property of a molecular graph.

In traditional GCNs [13], the connectivity of the graph is summarized into a binary adjacency matrix A_{binary} where the ij-th entry has a value of 1 if there is an edge connecting node i and node j; or otherwise has a value of 0. Based on A_{binary}, the spectral convolutions on graph are defined as the multiplication of a signal $x \in X$ with a filter g_θ parameterized by θ from matrix coefficients Θ:

$$g_\theta \star x = U g_\theta U^T x, \tag{1}$$

where U is the matrix of eigenvectors of the normalized graph Laplacian $L = I_n - D^{1/2} A_{binary} D^{1/2}$, where I is the identity matrix and D is the degree matrix of the fixed binary adjacency matrix A_{binary}.

However, when modeling molecular graphs, we need to pay attention to two properties. First, some bonds can be more important than others for a specific structure–activity relationship, which cannot be implemented by a binary adjacency matrix which amounts to equally summing up the information of all neighbors. Naturally different attention weights should be given to different

neighbors when aggregating node embeddings, in order to predict a graph property. For example, in the aromaticity view, the weights for aromatic bonds in any molecules can be closer than those assigned to non-aromatic bonds. Second, different molecular graphs have consistent bond energies (enthalpies) and bond lengths (interatomic distances) for the same type of bond, called edge consistency, which has been illustrated in the introduction. The bonds with the same bond type can be assigned with the same attention weight.

Although attention-based GCNs have been studied [41,44] which learn a dynamic and adaptive aggregation of the neighborhood, these methods such as GAT [44] determine the weight for an edge based on node similarity. They cannot handle the edge consistency which is important for molecules. In addition, most of the existing methods [12,35,41,44] including GAT are designed for a single large graph. It has the difficulty of designing an attention method that enforces edge consistency across multiple molecular graphs of different sizes and shapes. All edges with the same type across all molecular graphs should have the same attention weight.

In this paper, we propose a CEM mechanism, which not only assigns different attention weights to different neighbors when aggregating node embedding, but also enforces a constraint of edge consistency on all edges of the same interaction type to have the same attention weight cross different input molecular graphs. This CEM mechanism is able to reflect the above two properties of molecules.

In addition, we model a molecular graph as multi-view graphs to utilize the edge attributes. Each edge in molecular graph has multiple edge attributes as shown in Table 1, where each attribute has several discrete values. Each discrete value from an edge attribute represents an edge type or an atom-to-atom interaction. Hence, for each edge attribute, we can get a group of atom-to-atom interactions or edge types. In this paper, each attribute including a group of edge types is defined as one view. For each view, our CEM function C defined in the following section will be able to build the consistent edge constraint by assigning attention weights based on this group of edge types. Note that this group of edge types will be shared cross all molecules.

After that, each view is represented as a graph with a consistent edge attention matrix where edge weights are specified according to edge types. Since we have multiple discrete edge attributes, a molecular graph can be decomposed into multiple graphs with a group of consistent edge attention matrices $\mathbb{A} = \{A_0, A_1, \ldots, A_k\}$. Based on function C and multiple edge attributes $\mathbb{Z}$, a novel edge consistent constraint based multi-view spectral graph convolution is proposed to collectively aggregate information from graph structure.

First, let us consider the i-th edge attribute (i-th view) $Z_i \in \mathbb{R}^{d_i \times 1}$ where d_i is the number of discrete values of attribute i. The objective is to learn a function $F_{\Theta,M}$ and a function P_W minimizing the empirical loss $\mathbb{L}$ with an edge consistency constraint:

$$\min_{\substack{\Theta^1,\ldots,\Theta^L,W \\ M^1,\ldots,M^L}} \mathbb{L}\left(P\left(F^L \circ F^{L-1} \circ \ldots \circ F^1\left(X^1; A_i^1, \Theta^1\right); W\right), y\right)$$

$$\text{subject to } A_i^l = C\left(A_{binary}, Z_i; M^l\right), \forall 1 \leqslant l \leqslant L, X^1 = X,$$

where $C\left(A_{binary}, Z_i; M^l\right)$ is the consistent edge mapping (CEM) function, M^l includes all parameters of C in layer l. The function P includes the graph representation layer, fully connected layer and activation functions where W contains all parameters of P. The graph convolutional network F is used in a composite function denoted by $\circ$ so

$$F^{l+1} \circ F^l := F^{l+1}\left(F^l\left(X^l; A^l, \Theta^l\right); A^{l+1}, \Theta^{l+1}\right),$$

where Θ^l includes all parameters of F in layer l.

Second, we extend our model to a multi-view neural network so multiple edge attributes $\mathbb{Z} = \{Z_1, \ldots, Z_k\}$ can be utilized. In the multi-view model, our objective function can be written as:

$$\min_{\substack{\Theta^1,\ldots,\Theta^L,W \\ M^1,\ldots,M^L}} ; \mathbb{L}\left(P\left(F^L \circ F^{L-1} \circ \ldots \circ F^1\left(X^1; \mathbb{A}^1, \Theta^1\right); W\right), y\right)$$

$$\text{subject to } \mathbb{A}^l = \left\{A_0^l, A_1^l, \ldots, A_k^l\right\}, A_i^l = C\left(A_{binary}, Z_i; M_i^l\right), \forall 1 \leqslant l \leqslant L, X^1 = X,$$

where the parameters M^l and Θ^l can be defined as $M^l = \left\{M_1^1, \ldots, M_k^l\right\}$ and $\Theta^l = \left\{\Theta_1^1, \ldots, \Theta_k^l\right\}$ and F is the mapping represented by a multi-view graph convolutional network layer.

In summary, we define the molecular property prediction task as an edge consistency constraint based graph learning problem, which reserves the general consistency of the bond energies and bond lengths in various molecules.

4. The EAGCN model

The architecture of the proposed *EAGCN* model is shown in Fig. 1. *EAGCN* begins with multi-view spectral GCN layers where each layer has two steps: CEM and Spectral Graph Convolution. The output of multi-view spectral GCN layers is the node embeddings. Then a graph representation layer combines all node embeddings to obtain the graph embedding. We compare the two methods including simple sum and differential pooling [31] to integrate node embeddings into a graph representation. Then a fully connected layer with an activation function is used to predict the molecular property y based on the graph representation of the molecule.

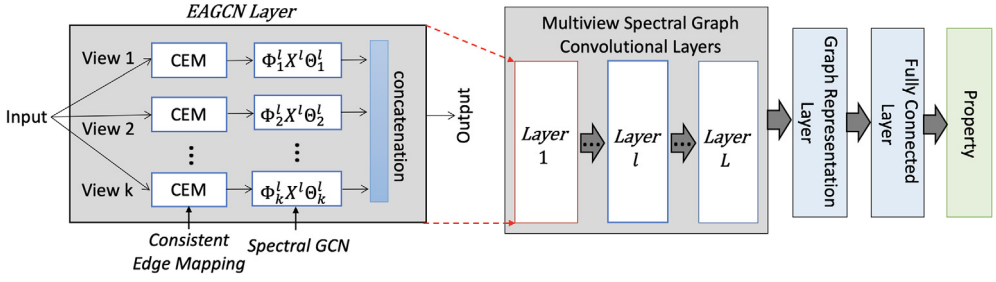
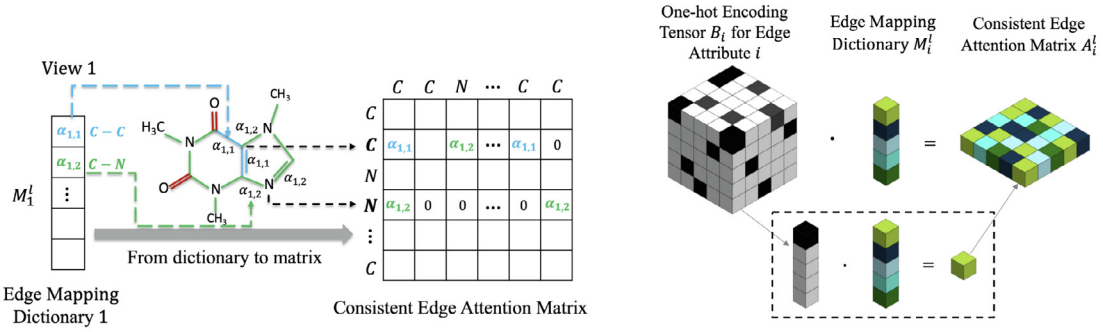


Fig. 1. The architecture of *EAGCN* model.



(a) Example of the edge mapping dictionary for view 1 in layer l.

(b) Example of consistent edge mapping: from one-hot encoding tensor to consistent edge attention matrix.

Fig. 2. The process of consistent edge mapping.

4.1. Consistent Edge Mapping (CEM)

The CEM mechanism generates a learnable consistent edge attention matrix A_i^l using the discrete values of the edge attribute i (view i) in layer l. If the edges have the same edge attribute no matter they are from the same molecule or different molecules, they are considered the same edge type. The same edge type receives the same attention weight, which will limit the parameter space to be optimized during GCN training. By assigning the same weight to all edges of the same type cross molecular graphs, edge consistency is automatically imposed. This consistency of assigning edge attention weights amounts to an "edge consistency constraint" imposed to the GCN architecture.

4.1.1. Creating edge mapping dictionaries

To better understand the importance of each edge type to the target property y, *EAGCN* learns the attention weights for graph edges at each neural network layer. However, it is difficult to assign weights with edge consistency for multiple graphs of different sizes and shapes.

In our paper, for each view i in layer l, we build an edge mapping dictionary M_i^l to satisfy the consistent edge attention constraint. For view i, if we have d_i discrete values for edge attribute i, edge mapping dictionary M_i^l is created with d_i learnable neural network parameters. In the dictionary, a discrete (attribute) value (i.e., an edge type) corresponds to a single parameter. For a view of a given molecular graph, our method will look up this dictionary to form an attention-based adjacency matrix according to the specific edge types in the graph. During the training of the GCN on this input graph, the values in the dictionary will be updated. For example, as shown in Fig. 2 (a), all edges in a view will search the edge mapping dictionary to get the attention weight. In Fig. 2 (a), we can see that all C–C edges will select the same $\alpha_{1,1}$ as the attention weight from M_1^l in layer l.

The attention weights in these dictionaries will be learned by our *EAGCN* model. We claim two points for the dictionaries: 1) the edge mapping dictionaries are defined for a dataset. Based on the edge attributes of a molecule, it is coded as weighted adjacency matrices of multiple views using these edge mapping dictionaries; 2) the dictionaries for edge attributes are used for all graphs in the dataset. Then a weighted adjacency matrix called consistent edge attention matrix is constructed for each view at each layer as described in the following subsection.

[1] One-hot encoding vector is a group of values among which the legal combinations of values are only those with a single value of 1 and all the others 0.

4.1.2. Consistent edge attention matrix

In layer l, we first create a binary one-hot encoding vector [1] $b_i(e) \in \mathbb{R}^{d_i}$ for each edge $e \in E$ in the view i, otherwise it will be a zero vector. Based on the vector $b_i(e)$, we will obtain a specific weight $\alpha_{i,j}^l$ for edge e by looking up the dictionary for this view:

$$\alpha_{i,j}^l = b_i(e)^T M_i^l, \tag{2}$$

where j denotes the j-th element in dictionary M_i^l, and i indexes the views. Note that $\alpha_{i,j}^l$ will be zero if $b_i(e)$ is a zero vector. As shown in Fig. 2 (b), all vectors form the binary tensor B_i for view i. By looking up all existing edges using the binary tensor, we get a newly generated consistent edge attention matrix A_i^l:

$$A_i^l = B_i \cdot M_i^l, \tag{3}$$

where "$\cdot$" is the lookup operation on the dictionary as shown in Fig. 2 (b). The global edge weights over all graphs preserve the consistent property of the relationship. In the experiment, we implement this mapping process using the 2D convolution. We consider the one-hot encoding tensor is an image with multiple channels. Here the channel size is the dimension d_i of the edge mapping dictionary. The kernel size we used is 1×1.

4.2. Multi-view spectral GCN

The previous section introduces the CEM to transform the edge types of graph signals to the edge consistency constraint. Based on this constraint, each molecular graph in layer l has been converted to a constrained graph signal, which consists of consistent edge adjacency matrices $\mathbb{A} = \left\{A_1^l, \ldots, A_k^l\right\}$ and feature matrix X. In this section, we propose a multi-view spectral graph convolution operation with a new spectral filter for the graph signal. The proposed multi-view GCN with the layer-wise propagation rule is presented in Section 4.2.1. Then Section 4.2.2 illustrates that this propagation rule is motivated via a new parameterization of the first order Chebyshev approximation of graph Laplacian.

4.2.1. The proposed graph convolution

There are multiple consistent edge attention weights (each corresponding to a view or edge attribute) for each edge in each layer. The graph convolution aggregates the node information over all first-order neighbors from each view. In layer l, the new spectral graph convolution for the constrained graph signal in view i is defined as:

$$X_i^{l+1} = \sigma\left(\Phi_i^l X^l \Theta_i^l\right) = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A}_i^l \tilde{D}^{-\frac{1}{2}} X^l \Theta_i^l\right), \tag{4}$$

where $\tilde{A}_i^l = A_i^l + r_i I$, $\tilde{D}$ is the corresponding normalized degree matrix of $\tilde{A}_i^l$, the ratio r_i is the weight of the self-loop edges in view i, σ is an activation function. Then the outputs from multiple views are concatenated:

$$X^{l+1} = Concate\left(X_1^l, X_2^l, \ldots, X_k^l\right), \tag{5}$$

where the X^{l+l} is used as the input of next layer. Since the parameter matrix Θ_i^{l+1} from next convolutional layer provides different neural network parameters for multiple views, the contributions of different views are learned by our model.

In each view, the spectral GCN layer aggregates the features of each node with all of its neighbors according to attention weights which reflect the "strengths" of node-to-node interactions for one edge attribute. By multi-view spectral graph convolutions, the proposed model explores the multiple types of "strengths" between two nodes in a molecule.

We summarize the computational steps when training layer l of the proposed GCN in Algorithm 1. A normalized consistent edge attention matrix Φ_i^l is calculated for each view i. The different edge types in different views will derive particular meaningful node embeddings, which are concatenated to form a complete node feature matrix. In the study of chemical compounds, such collection gives different perspectives of atomic interaction and strength of influence.

Algorithm 1: Consistent Edge-Aware Multi-View Spectral Graph Convolutional Layer l

Input: View $i \in [1, k]$; Layer l; Binary Tensors $\boldsymbol{B}_i$; Node Feature Matrix $\boldsymbol{X}^l$
Trainable Parameters: Edge Mapping Dictionaries $\boldsymbol{M}_i^l$; Self-attention Weights $\boldsymbol{r}_i^l$; Parameters Matrices $\Theta_i^l, i = 1, 2, \ldots, k$
Multi-View Spectral Graph Convolutions:
for $i = 1$ to k **do**
  $\boldsymbol{A}_i^l = \boldsymbol{B}_i \cdot \boldsymbol{M}_i^l$ – Consistent Edge Mapping
  $\tilde{\boldsymbol{A}}_i^l = \boldsymbol{A}_i^l + \boldsymbol{r}_i^l I$
  $\Phi_i^l = \left(\tilde{\boldsymbol{D}}_i^l\right)^{-1/2} \tilde{\boldsymbol{A}}_i^l \left(\tilde{\boldsymbol{D}}_i^l\right)^{-1/2}$ where $\tilde{\boldsymbol{D}}_i^l$ = diag(row-sum $(\tilde{\boldsymbol{A}}_i^l)$), that is the Diagonal Degree Matrix.
  $\boldsymbol{X}_i^l = \Phi_i^l \boldsymbol{X}^l \Theta_i^l$
end for
Output: $\boldsymbol{X}^{l+1} = Concate\left(\boldsymbol{X}_1^l, \boldsymbol{X}_2^l, \ldots, \boldsymbol{X}_k^l\right)$

Graph Invariance and Varying Graph Size. The consistent edge attention matrices imply the multiple strengths of connection and interaction between nodes. The attention weights are conditioned on edge mapping dictionaries instead of the neighborhood order. These edge mapping dictionaries result in a homogeneous view of local graph neighborhoods. These weights are shared over all molecular graphs, which enables us to extract the locally stationary property of the input data by revealing local features that are shared across all graphs. Hence, our model has been designed to extract invariant features by the consistent edge attention, and provides a solution to solve the graph invariance problem [24]. In addition, for each edge attention layer, the number of attention parameters equals the number of different values in an edge attribute rather than the actual number of edges in a graph. Thus, the number of parameters in *EAGCN* is much smaller than other GCNs with a real-valued adjacency matrix and insensitive to the varying graph size. The varying size has no influence on the complexity of the model.

Interaction between Substructures. From Eq. (4), for each node, the information is exchanged only with its neighbors within a graph convolution layer. However, if we consider such information propagation from layer to layer, the attentions at the higher layers learn the interactions of substructures. The attention weights in layer 1 represent the atomic interaction between two atoms in the first order neighbors, while the attention weights in layer 2 propagate the neighborhood information to the second-order neighbors, and so learn the interaction between substructures. The attention weights are thus capable of characterizing substructure within different scopes to learn the connection information.

4.2.2. Spectral analysis

The graph convolutional operation presented in Section 4.2.1 is motivated via the multiplication of a constrained graph signal $\mathbb{A}$ with a new spectral filter. We explain the new spectral filter on the feature matrix X^l and consistent edge adjacency matrix A_i^l in view i and layer l. Because the spectral analysis is the same across layers and views, we omit the layer superscript and view subscript in this subsection.

The Laplacian of Consistent Edge Adjacency Matrix. An essential tool in spectral graph analysis is the graph Laplacian matrix [37]. We get the normalized graph Laplacian matrix for consistent edge attention matrix, defined as $L = I - D^{-1/2} A D^{-1/2}$, where I is the identity matrix, D is a diagonal degree matrix which contains information about the degree of each vertex. We first show that the graph Laplacian matrix L of the edge adjacency matrix A satisfies the following properties, and the proof can be referenced in Appendix A:

1. For every vector $x \in \mathbb{R}^n$, we have $x^T L x \geqslant 0$, that means L is a positive semi-definite matrix.
2. L is a symmetric matrix with n non-negative, real-valued eigenvalues (given L is positive semi-definite): $0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n = \lambda_{max} \leqslant 2$.

Spectral Convolutions on Graph Signals. Based on spectral theory, a positive semi-definite matrix can always be diagonalized using a basis of eigenvectors, so it can be written as $L = U \Lambda U^T$ for a unitary matrix U and a diagonal matrix $\Lambda = diag(\lambda_i)$. If we multiple the L matrix s times as $L^s = \left(U \Lambda U^T\right)^s = U \Lambda^s U^T$, which is the basis of s-localized convolution in a graph convolution analysis.

The convolution operator on graph in the Fourier domain is defined as $x * y = U\left(\left(U^T x\right) \odot \left(U^T y\right)\right)$, where $\odot$ is the element-wise Hadamard product. The graph Fourier transform of a graph signal x is defined as $\hat{x} = U^T x \in \mathbb{R}^n$, and the $U^T y$ and $\odot$ operations perform the filtering of the $U^T x$ in the Fourier domain. Thus we can define a signal x with a filter g_θ in the Fourier domain as:

$$y = g_\theta(L)x = g_\theta\left(U \Lambda U^T\right)x = U g_\theta(\Lambda) U^T x, \tag{6}$$

where θ is a vector of Fourier coefficients. However, the cost of the filtering operation on signal x is as high as $O(n^2)$ because of the multiplication with the Fourier basis U. To alleviate this issue, few approaches [36,52] are proposed to parameterize $g_\theta(L)$ as a polynomial function, such as the Chebyshev polynomials.

The Chebyshev polynomials form a recursive sequence defined as $T_s(x) = 2xT_{s-1}(x) - T_{s-2}(x)$, where $s = 2, 3, \cdots$ is an index, and $T_0(x) = 1, T_1(x) = x$. The operator $g_\theta(L)$ can then be computed recursively from Laplacian L through a Chebushev approximation.

We can derive the S-th order polynomial approximation of the convolution operator $g_\theta(L)x$ in terms of the eigen-values Λ of L:

$$g_\theta(\Lambda) \approx \sum_{s=0}^{S} \theta_s T_s\left(\tilde{\Lambda}\right), \tag{7}$$

where we use the re-scaled eigenvalue matrix $\tilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I$ to help make sure that the eigenvalues used in the polynomial approximation lie in the range $[-1, 1]$, where the $\lambda_{max} \approx 2$. Correspondingly, we have the $\tilde{L} = L - I = -D^{-1/2}AD^{-1/2}$.

Now, similar to all other GCNs, we examine the first order approximation with $S = 1$. The Chebyshev approximation is linear with respect to the graph Laplacian spectrum, specifically:

$$\begin{aligned} y &= g_\theta(L)x = Ug_\theta(\Lambda)U^T x \\ &\approx U\left(\theta_0 + \theta_1\tilde{\Lambda}\right)U^T x = \theta_0 x + \theta_1 \tilde{L}x \\ &= \theta_0 \cdot I \cdot x + \theta_1\left(-D^{-1/2}AD^{-1/2}\right)x. \end{aligned}$$

The spectral filter in traditional GCNs [13] is also derived from the first order Chebyshev approximation relying on these two coefficients - θ_0 and θ_1. However, these two coefficients are enforced to satisfy $\theta_0 = -\theta_1$, which thus results in only a single free parameter θ.

The New Spectral Filter. In this paper, we relax the "$\theta_0 = -\theta_1$" requirement to derive a more general and effective parameterization of the GCN network. We employ a parameter θ and a real-valued ratio r to satisfy the following conditions:

$$\begin{cases} \theta_0 = r\theta \\ \theta_1 = -\theta, \end{cases}$$

where ratio r specifies the weight of the self-loop edges which will be proved in the coming paragraph. Then the expression will come out as:

$$\begin{aligned} y &\approx r\theta \cdot I \cdot x + \theta\left(D^{-1/2}AD^{-1/2}\right)x \\ &= \theta\left(rI + D^{-1/2}AD^{-1/2}\right)x. \end{aligned}$$

Different from the traditional spectral analysis of $I + D^{-1/2}AD^{-1/2}$, the weighted graph we will learn at each layer includes a weighted self-edge to each node itself. Hence, the eigenvalues will be nested in the range $[r-1, r+1]$. Note that when the ratio is 1, our method is degenerated into the traditional graph convolution. In order to leave the eigenvalues in the range $[0, 1]$ to ensure a stable deep neural network, we re-normalize:

$$y \approx \theta\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\right)x, \tag{8}$$

where $\tilde{A} = A + rI$, and $\tilde{D}$ is the corresponding normalized degree matrix of $\tilde{A}$. An interesting observation is that the ratio r can be considered as the weight that the self-loop edges play in the convolution as illustrated in Algorithm 1. By adding this parameter into the GCN, each node has the ability to control the importance of its original information and features. In summary, the propagation rule in our GCN based on the consistent edge attention matrix is derived based on the first-order Chebyshev approximation with more plausible assumptions and reasonable parameterization than early works (e.g., [13]).

Putting all these θ parameters together for all graph signals in $X \in \mathbb{R}^{n\times f}$ with f input channels yields a filter parameter matrix $\Theta \in \mathbb{R}^{f\times f'}$ with f' filters, hence we get:

$$Y = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta = \Phi X\Theta, \tag{9}$$

where $\Phi = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$. We thus derive the convolved signal matrix $Y \in \mathbb{R}^{n\times f'}$. If the input graph signal is X^l, the output Y of the proposed method will be X^{l+1}. When we extend our analysis to multi-view graph convolution, a set of ratio parameters $\{r_1, \ldots, r_k\}$ are used in our model. These parameters provide more expressive network models than the traditional GCN to benefit the property prediction.

4.3. The graph representation layer

After the node representations have been generated in the early layers of the proposed GCN, another step is to compute the graph representation using these node representations. Two aggregation methods are provided and we compare them in the Experiments section.

4.3.1. Simple sum

A simple way to obtain the graph representation is summing up all node embeddings or representations:

$$X^{l+1} = Sum\left(x_1^l, x_2^l, \ldots, x_n^l\right) \in \mathbb{R}^{1\times f_l},$$

where x_i^l means the representation of node i in layer l, n is the number of node, and f_l is the dimension of node representation in layer l.

4.3.2. Differentiable pooling

The differentiable pooling (Diffpool) [31] is proposed to learn the hierarchical representation of a graph. Diffpool designs a graph pooling neural network to generate a cluster assignment matrix for nodes:

$$S^l = softmax\left(GCN_1^l\left(A^l, X^l\right)\right) \in R^{n_l\times n_{l+1}},$$

where the softmax function is applied in a row-wise fashion, the inputs are the feature matrix X^l and cluster adjacency matrix A^l, and GCN_1^l is the traditional graph convolutional network [13] at layer l. In the first layer of Diffpool, $n_l = n$. Because we have multiple consistent attention matrices, the input of first differential pooling layer is $A^l = \sum_{i=1}^{k}\tilde{A}_{att,i}^{l-1}$. Then the new cluster adjacency matrix and the feature (representation) matrix of cluster nodes can be obtained using another GCN_2^l as follows:

$$X^{l+1} = \left(S^l\right)^T GCN_2^l\left(A^l, X^l\right),$$

$$A^{l+1} = \left(S^l\right)^T A^l S^l.$$

This final output embedding is the graph representation $X \in \mathbb{R}^{1\times f_G}$where f_G is the dimension of the graph representation.

5. Experiments

We perform an extensive empirical analysis based on five benchmark datasets with respect to both the regression and classification tasks. We also make efforts to understand the GCN-derived molecular representation and interpret the attention weights that are learned by our model.

5.1. Benchmark datasets

Five benchmark datasets [22,53] (Tox21, Freesolv, Lipophilicity, eSOL and CPAL) are utilized in this study to evaluate the predictive performance of the *EAGCN* model.

Table 2
Regression results for the three benchmark datasets. EAGCN[1] is the $EAGCN_{sum}$ and EAGCN[2] is the $EAGCN_{diffpool}$.

Method	Regression (RMSE)					
	Lipo		Freesolv		eSOL	
	Valid	Test	Valid	Test	Valid	Test
RF	0.85 ± 0.02	0.85 ± 0.03	2.31 ± 0.82	1.62 ± 0.14	1.27 ± 0.07	1.18 ± 0.11
Weave	0.85 ± 0.10	0.88 ± 0.09	1.63 ± 0.25	1.37 ± 0.19	0.90 ± 0.08	0.85 ± 0.03
MPNN	0.81 ± 0.05	0.80 ± 0.05	1.26 ± 0.42	1.09 ± 0.19	0.66 ± 0.07	0.67 ± 0.06
GAT	0.85 ± 0.02	0.84 ± 0.01	1.73 ± 0.25	1.45 ± 0.25	0.86 ± 0.00	1.00 ± 0.20
GCN	0.60 ± 0.03	0.64 ± 0.03	1.24 ± 0.26	0.94 ± 0.10	0.76 ± 0.08	0.80 ± 0.17
EAGCN[1]	$\mathbf{0.56 \pm 0.03}$	$\mathbf{0.60 \pm 0.03}$	$\mathbf{1.07 \pm 0.28}$	$\mathbf{0.81 \pm 0.04}$	$\mathbf{0.54 \pm 0.06}$	$\mathbf{0.61 \pm 0.10}$
EAGCN[2]	0.61 ± 0.08	0.63 ± 0.04	1.14 ± 0.27	0.86 ± 0.09	0.59 ± 0.06	0.65 ± 0.06

Tox21. The original Tox21 data comes from the Toxicology in the 21st century research initiative. It contains 7831 environmental compounds and drugs as well as their biological outcomes of 12 pathway assays.

Freesolv. Freesolv is a database of experimental and calculate hydration free energies for small neutral molecules in water, along with molecular structures, input files, references, and annotations [54]. It includes a set of 642 neutral molecules which are mostly fragment-like.

Lipophilicity (Lipo). Lipophilicity, curated from ChEMBL database, provides experimental results of octanol/water distribution coefficient of 4200 compounds.

eSOL. eSOL is a database on the solubility of entire ensemble E.-coli proteins [55] individually synthesized by PURE system that is chaperone free.

CPAL. The "Cytotoxic Profiling of Annotated Libraries Using Quantitative High-Throughput Screening" (CPAL) dataset is provided by Pubchem. The CPAL data can be download from Pubchem website[2].

Four of these datasets except CPAL are downloaded from the MoleculeNet website[3] that hold various benchmark datasets for machine learning studies of molecules. In our experiments, we only use molecules with common atom types, e.g. boron, carbon, nitrogen, oxygen, halogen, phosphorus, sulfur.

5.2. Experimental setup

Our experiments are designed to evaluate the QSAR prediction on the standard supervised classification and regression tasks. We design our experiments with the goals of verifying the improvement of our method over several baseline methods, such as the classic GCN [13] and providing more in-depth interpretation of the molecular representation and atom embeddings.

The node features and edge attributes are extracted using the RDKit[4], an open source cheminformatics package. RDKit also converts SMILES strings into RDKit "mol" format, which contains the molecular structure information used to build the molecular graphs. Here we ignore the SMILES samples whose structural graphs have no edge. The edge attributes [29] are shown in the Table 1. When we build the edge mapping dictionaries for different atom-pair types, we set a threshold on the frequency of atom pair types for each dataset. For the atom pair types whose frequencies are lower than the threshold, we give a fixed attention weight for them in the dictionary. Each dataset is randomly split into three sets: training (80%), validation (10%), and test (10%). For each experiment, five independent runs with different random seeds are performed. The number of multi-view spectral graph convolutional layers is 4. For fair comparison, the number of layers for GCN baseline is also 4. In addition, when we use the Diffpool aggregation layer to obtain graph representation, the number of such layers is 2. All results present here are the average of five runs, with standard deviations listed. We use the adaptive moment (Adam) algorithm [56] to train the deep GCN model and set the learning rate to 0.0001 for freesolv and tox21 datasets and 0.001 for other datasets.

5.3. Molecular property prediction

We compare our model against several benchmark methods which are shown in MoleculeNet [22]. First, the Kernel-SVM and Random Forests (RF) which are the most widely used machine learning methods for molecules are compared in our experiments. Second, the classic GCN [13] is the baseline for the comparison. In addition, we also compare with three other models which leverage the edge attributes. Weave model [24] is similar to our graph convolutions. The weave featurization encodes both the local chemical environment and connectivity of atoms in a molecule, and calculates a feature vector for each pair of atoms in the molecule. Message passing neural network (MPNN) [23] is a generalized model of GCN, which learns a message passing algorithm and aggregation procedure to compute a function of their entire input graph. GAT is the benchmark of the attention-based GCN, which determines the weight for an edge based on the similarity of the two nodes.

5.3.1. Regression analysis

Solubility and lipophilicity are two basic physical chemistry properties that are important for understanding how molecules interact with solvents. Table 2 reports root-mean-squared-error (RMSE) results of five different comparison models and our *EAGCN* model. The comparison between $EAGCN_{sum}$ and $EAGCN_{diffpool}$ shows that the simple sum operation in graph representation layer has slightly better performance than the complex Diffpool operation. We believe that it may be partially due to the Diffpool method [31] that was proposed to handle larger graphs. Large graphs may be more suitable to learn the hierarchical graph representation. When the inputs are the small molecular graphs with different sizes and shapes, simple sum operation seems outperforming in obtaining a graph representation.

We compare the $EAGCN_{sum}$ with the best baseline models in [22] on the regression task. In Lipophilicity dataset, $EAGCN_{sum}$ achieves about 6.7% and 6.3% performance increases in comparison with the GCN which is the best baseline on the validation and test datasets. For the Freesolv dataset, $EAGCN_{sum}$ improves over the GCN by a margin of 13.7% and 13.8% in validation and test. For the eSOL dataset, $EAGCN_{sum}$ improves over MPNN by a margin of 18.2% on validation datasets and by a margin of 9.0% on test datasets. *EAGCN* achieves the best performance, which demonstrates that using the consistent edge mapping improves the molecular property prediction.

EAGCN shows strong validation and test results on all datasets. Given the FreeSolv dataset contains only around 600 compounds,

[2] https://pubchem.ncbi.nlm.nih.gov/bioassay/1296008
[3] http://moleculenet.ai/
[4] http://www.rdkit.org/

Table 3
Classification results for the two benchmark datasets. EAGCN[1] is the $EAGCN_{sum}$ and EAGCN[2] is the $EAGCN_{diffpool}$.

Method	Classification (ROC-AUC)			
	Tox21		CPAL	
	Valid	Test	Valid	Test
Kernel-SVM	0.77 ± 0.01	0.77 ± 0.02	0.79 ± 0.02	0.76 ± 0.06
RF	0.76 ± 0.01	0.77 ± 0.01	0.77 ± 0.05	0.74 ± 0.06
Weave	0.81 ± 0.01	0.83 ± 0.01	0.78 ± 0.06	0.77 ± 0.08
MPNN	0.80 ± 0.02	0.82 ± 0.02	0.83 ± 0.04	0.78 ± 0.07
GAT	0.84 ± 0.02	0.84 ± 0.02	0.81 ± 0.04	0.83 ± 0.05
GCN	0.83 ± 0.01	0.84 ± 0.01	0.88 ± 0.02	0.80 ± 0.07
EAGCN[1]	$\mathbf{0.85 \pm 0.01}$	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.90 \pm 0.02}$	$\mathbf{0.85 \pm 0.06}$
EAGCN[2]	0.85 ± 0.01	0.85 ± 0.01	0.87 ± 0.02	0.83 ± 0.07

our model still reach excellent performance by training on this limited sample. From the table, graph-based methods, such as *EAGCN* and graph convolutional model, all exhibit significant boosts over tasks, indicating the advantages of learnable featurizations. In these three datasets, data-driven methods outperform physical algorithms with moderate amounts of data. These results suggest that graph convolution based approaches are powerful alternatives to QSAR analysis.

5.3.2. Classification analysis

Table 3 reports the Receiver Operating Characteristic curve and the Area Under the Curve (ROC-AUC) results of six different baseline models on Tox21 and CPAL datasets. On the Tox21 dataset, the GAT baseline model achieves the best performance among all baselines on the test and validation datasets. $EAGCN_{sum}$ improves from the performance of GAT by a margin of 1.2% and 2.4% on the validation and test datasets. For the CPAL dataset, $EAGCN_{sum}$ model improves over the GCN by a margin of 2.3% on validation dataset and achieves about 2.4% improvement than GAT on the test dataset. These results show that $EAGCN_{sum}$ consistently achieves reasonable and excellent performance in predicting QSAR properties. By referencing the last two rows in Table 3, $EAGCN_{sum}$ performance is still slightly better than molecular property prediction task comparing to $EAGCN_{diffpool}$, which is consistent with the regression task. There are many applications which can benefit from our model. For example, the classification model created using the Tox21 dataset can be further utilized to identify any new compounds with potential liability and prioritize specific compounds for more extensive toxicological evaluation.

5.3.3. The t-test results

In Table 4, the mean and standard deviation of model performance are computed according to the five trials using different data splits. In each trial, all models are compared on the basis of the same training, validation, and test set under the same random seed (i.e., the same split). To compare if the two different models are significantly different, we do a paired t-test between the prediction accuracy on both validation set and test set. The null hypothesis H_0 is that the two models are the same. In Table 5, the p-values in the column of "$EAGCN_{sum}$ vs GCN" are all less than or equal to 0.05. Thus, our $EAGCN_{sum}$ model performed significantly differently from the GCN model at a significance level of $\alpha = 0.05$. In the column of "$EAGCN_{sum}$ vs $EAGCN_{diffpool}$", the p-values indicate that the two different graph representation methods have no significant difference (i.e., failure to reject H_0).

Table 4
The paired t-test for model comparison.

Models		p-value	
		$EAGCN_{sum}$ vs GCN	$EAGCN_{sum}$ vs $EAGCN_{diffpool}$
Lipo	Validation	0.0373	0.3105
	Testing	0.0232	0.1992
Freesolv	Validation	0.0297	0.1966
	Testing	0.0247	0.1010
eSOL	Validation	0.0265	0.0562
	Testing	0.0106	0.3697
Tox21	Validation	0.0002	0.0921
	Testing	0.0156	0.0719
CPAL	Validation	0.0129	0.0923
	Testing	0.0305	0.2862

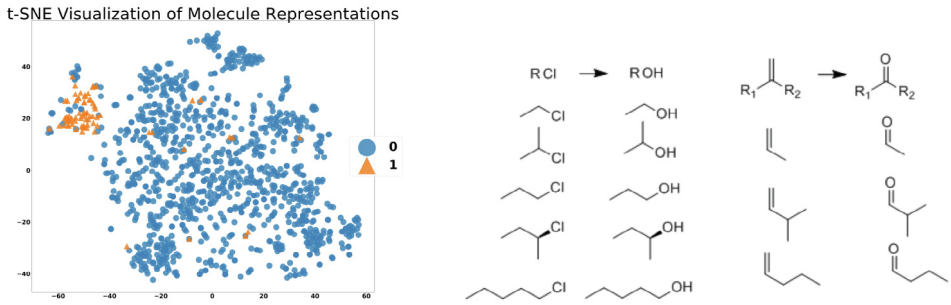
Table 5
Sample molecules with similar structure but different property in the Freesolv dataset where "=" is double bond and "#" is triple bond.

Name	SMILES	Label	Prediction
1-chloro-2-methyl-benzene	**Cc1ccccc1Cl**	−1.14	-1.14
2-methylaniline	**Cc1ccccc1N**	−5.53	−5.53
o-cresol	**Cc1ccccc1O**	−5.9	−5.69
pent-1-yne	**CCCC#C**	0.01	0.10
pent-1-ene	**CCCC = C**	1.68	1.49
pent-1-ene	CCCC = C	1.68	1.49
butanal	CCCC = O	−3.18	−3.21
2-chloro-2-methyl-propane	CC(C)(C) Cl	1.09	1.43
2-methylpropan-2-ol	CC(C)(C) O	-4.47	−4.35
isobutane	CC(C) C	2.3	2.17
2-bromopropane	CC(C) Br	−0.48	−0.40
1-ethyl-2-methylbenzene	CCc1ccccc1C	−0.85	−0.91
2-ethylphenol	CCc1ccccc1O	-5.66	−5.62
pent-1-yne	CCCC#C	0.01	0.10
butanenitrile	CCCC#N	-3.64	−3.78
hex-1-ene	CCCCC = C	1.58	1.70
pentanal	CCCCC = O	-3.03	−2.95

5.4. Molecular graph embedding analysis

We employ the widely-used t-SNE program [57] to reduce the dimension of the molecular graph representations and visualize the molecules in the CPAL dataset in the projected 2-dimensional (2D) space in Fig. 3(a). The t-SNE algorithm is a nonlinear dimensionality reduction technique to embed the high-dimensional atom vector representation to a two-dimensional space. It constructs the probability distribution over data points in both original high-dimensional space and the reduced low-dimensional space, and then minimizes the Kullback–Leibler divergence between the two distributions. The CPAL molecules are labeled with respect to toxicity of either 1 or 0. The striking observation is that most molecules labeled differently can be clearly separated. The points labeled by 1 (triangle) are restricted to the top-left corner of the projection plane, and the molecules with label 0 are scattered over a large area. Because this dataset is unbalanced in terms of the 1 and 0 labels, so we have down-sampled the molecules with label 0.

Since many molecular graphs have similar structure but different property values, it is a challenge to predict their property



(a) Visualization of molecular graph embeddings in CPAL. (b) Matched molecular pair (MMP) examples.

Fig. 3. Graph representation visualization and matched molecular pair examples.

accurately. Fig. 3(b) shows several exemplar molecules that differ only by a single chemical transformation, but their properties are quite different. Table 5 shows more examples with property values where only one atom of the first three molecules is different. In addition, only one bond between the two molecules in the second block is different. Different bond orders and atoms do lead to different property values. The remaining rows of Table 5 contain more examples. After comparing the true property and prediction values, we see that our predictions for these cases are rather accurate even when they have similar structures.

The potential relationships cross molecules are very similar to the transformation between the word vectors in Natural Language Processing (NLP). The word embeddings often preserve the relations in the projected 2D space. For example, the relationship between male and female [58] is automatically learned. The "Queen" embedding can be derived from "King - Man + Woman". In the field of chemoinformatics, chemists are often interested in comparing properties of two molecules that differ only by a single chemical transformation, such as the substitution of a hydrogen atom by a chlorine atom. Such pairs of compounds are usually referred as matched molecular pairs (MMP), as demonstrated by Fig. 3(b). We are interested in assessing whether our atom embeddings can capture the singularity relationship among certain matched molecule pairs in the investigated dataset and further providing visual interpretation. We thus examine the 2D PCA plots for several subsets of matched molecule pairs in Fig. 4. Similar to the vector analyses in Word2Vec, the vector transformations between each matched pair are mapped to the 2D PCA plot.

As illustrated in Fig. 4, we obtain a series of relatively parallel vector transformation lines within each subset of matched molecule pairs, which shows that our *EAGCN* model may capture the implicit relationships of each molecular pair and such relationships can be useful to predict the relevant chemical properties of related analogs. For instance, when chloro substitutions are replaced with the hydroxy group in the first MMP example, the resulting transformation vectors are almost parallel and all point toward the same direction, which imply that these similar structural changes lead to similar changing effects on solubility property.

5.5. Atom subtype analysis

In our study, atom (sub) types of molecules are determined prior to the GCN modeling. Atom types are classified in Table 6 based on their elements, atomic connectivity and hybridization states. The atomic representation vectors are collected from the output of the last multi-view spectral graph convolutional layer in our model. We visualize these high-level atom representations learned by the GCN nets using t-SNE.

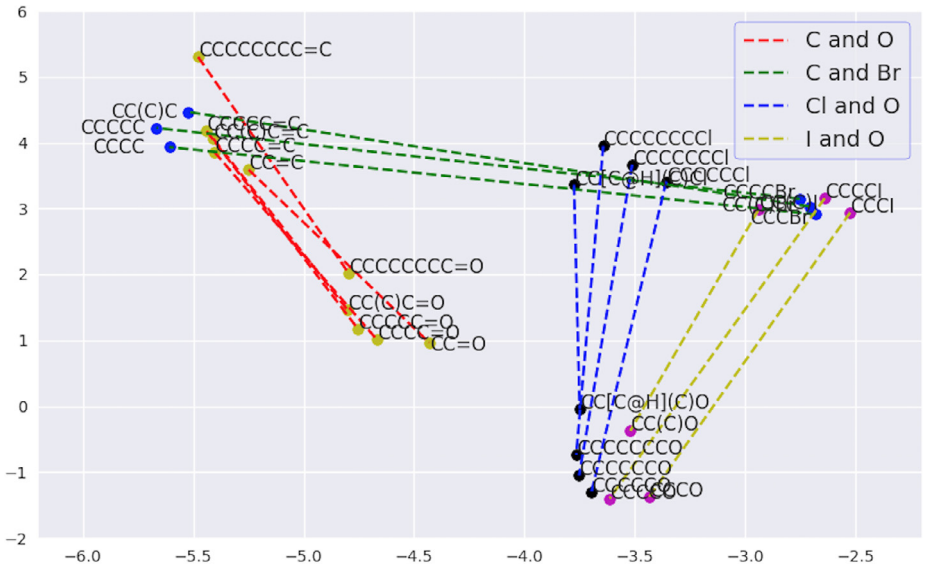
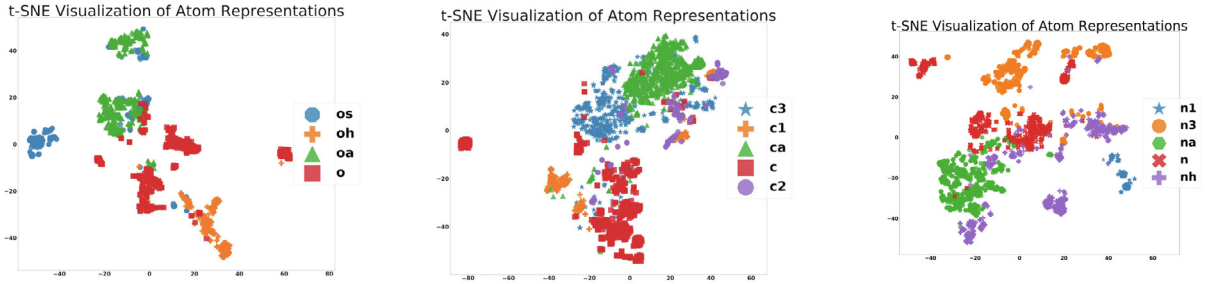


Fig. 4. Visualization of molecular representation using PCA in Freesolv dataset.

Table 6
Atom Subtype Definition.

Atom	Subtype	Definition
O	O	carbonyl oxygen
	Oa	sp2-hybridized oxygen in the aromatic ring
	Oh	sp3-hybridized oxygen in alcohol
	Os	sp3-hybridized oxygen in ether or ester
C	C	carboxylate, carbonyl and thion carbon
	C1	sp-hybridized carbon
	C2	sp2-hybridized noaromatic carbon with one substitution
	C3	sp3-hybridized carbon
	Ca	aromatic carbon
N	N	sp2-hybridized with 3 substituents and sp2-hybridized nitrogen in base NH2 group
	N1	sp1-hybridized nitrogen
	N3	sp3-hybridized nitrogen
	Na	sp2-hybridized aromatic nitrogen
	Nh	sp2-hybridized nitrogen in amide group



(a) Visualization of *O* subtypes. (b) Visualization of *C* subtypes. (c) Visualization of *N* subtypes.

Fig. 5. Visualizations of atom embeddings using t-SNE in Lipo. Points are colored by atom subtypes defined by Chemist.

As shown in Fig. 5, vector representations of various oxygen and carbon atom subtypes from *EAGCN* model in the Lipo dataset are mapped into the 2D t-SNE visualization plots. In each t-SNE plot, projected atom type representations are color coded with respect to atom subtypes specified in Table 6. For instance, in Fig. 5(s), depending on their chemical bonding environment, oxygen (*O*) atoms within the molecules of the Lipo dataset can be classified as the following four subtypes: sp3-hybridized oxygen in alcohol (oh), sp2-hybridized oxygen in the aromatic ring (oa), carbonyl oxygen (o), sp3-hybridized oxygen in ether or ester (os). Interestingly, the different atom subtype representations learned by the proposed GCN model are relatively well clustered in the

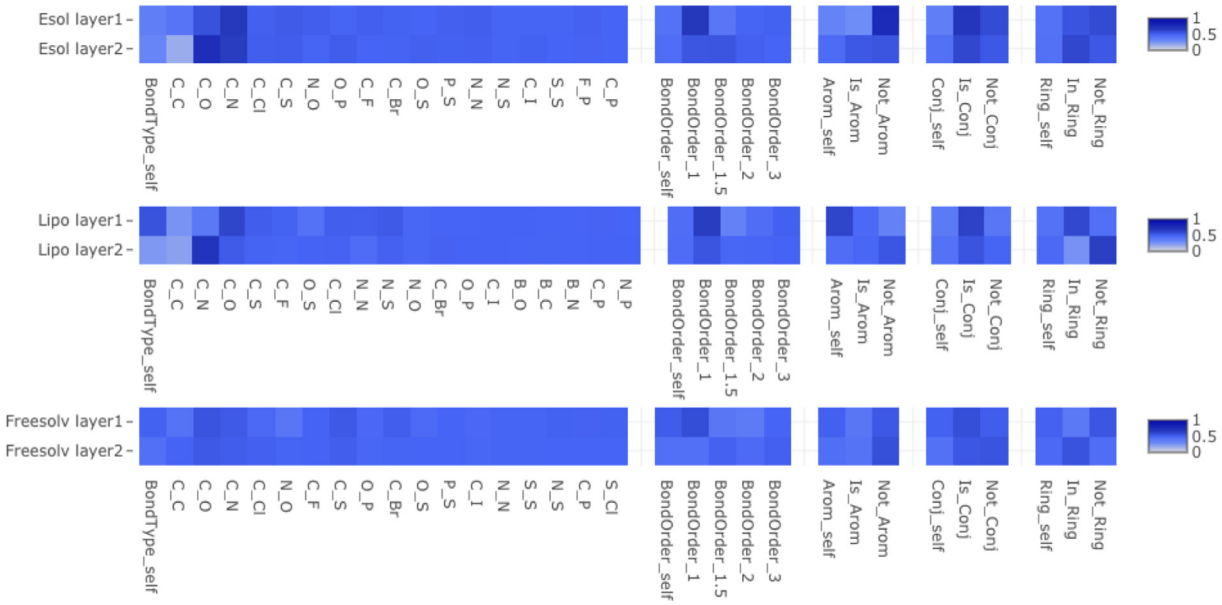


Fig. 6. Visualization of the learned edge attention weights. The labels are the edge types. The self-loop weights r_i^i's are labeled with "_self".

Table 7
Ablation test on single-view settings. The views correspond to the attributes in Table 1.

Method	Regression (RMSE)			
	Lipo		Freesolv	
	Valid	Test	Valid	Test
$EAGCN_{multiview}$	$\mathbf{0.56 \pm 0.03}$	$\mathbf{0.60 \pm 0.03}$	$\mathbf{1.07 \pm 0.28}$	$\mathbf{0.81 \pm 0.04}$
$EAGCN_{view1}$	0.59 ± 0.03	0.63 ± 0.04	1.15 ± 0.18	0.94 ± 0.11
$EAGCN_{view2}$	0.59 ± 0.03	0.62 ± 0.03	1.18 ± 0.27	0.96 ± 0.08
$EAGCN_{view3}$	0.59 ± 0.03	0.64 ± 0.03	1.13 ± 0.23	0.93 ± 0.13
$EAGCN_{view4}$	0.60 ± 0.04	0.63 ± 0.04	1.14 ± 0.23	0.91 ± 0.16
$EAGCN_{view5}$	0.59 ± 0.03	0.63 ± 0.03	1.18 ± 0.21	0.96 ± 0.11

t-SNE plot, which is consistent with the predefined atom subtypes based on the chemical intuition. We observe similar effects for the carbon (C) and nitrogen (N) atom subtypes, as demonstrated by Fig. 5(b) and 5(c). Hence, the atom embeddings of the *EAGCN* model are apparently able to pick up the atom type and subtype information.

5.6. Interpretation of attention weights

From the physical chemistry perspective, our edge mapping dictionaries learn to characterize multiple strengths of influence from neighboring atoms, which may offer some insights into atomic interactions. In Fig. 6, the weights stored in the edge mapping dictionaries are visualized by heatmaps. The darkness of a block corresponds to the magnitude of attention weights. The attention weights in Layer 1 represent the atomic interaction between two atoms, while the attention weights in Layer 2 model the interaction between two substructures centered at two atoms. Hence the attention weights in different layers have different values. In Fig. 6, both $C-O$ and $C-N$ attention weights in the first layer of the GCN model for the eSOL and Freesolv datasets have higher attention values. This is consistent with the analysis in Table 5, which shows that O and N are highly related with the solubility property. For edge mapping dictionaries of the Lipo dataset, the $C-N$ has a large weight in the second layer, which means the substructure around C has a significant strength of influence to the substructure around N for the property. This analysis of attention weights helps us understand the contributions of different interaction types to the property prediction.

5.7. Ablation tests

In order to analyze the importance of each edge attribute, we show the results of ablation tests in Table 7 including both multview and single-view settings. The comparison clearly shows: 1) the *EAGCN* model using multiple data views has better performance than that learned using single-view data only; 2) *EAGCN* under the single-view settings also achieves competitive performance comparing to baselines; 3) different views or attributes have different contributions and levels of importance for the property prediction. For example, as shown in the Freesolv test results, view 4 ("Conjugation" view in Table 1) is more important to the solubility prediction of molecules. We observe that different attributes influence differently on different properties.

6. Conclusion

We have introduced a consistent Edge-Aware multi-view spectral GCN (*EAGCN*), which models the graph connectivity in multiple views by parameterizing the graph deep network with consistent edge attention weights for graph representation and molecular property prediction. Because attention weights from an edge mapping dictionary are shared across all molecular graphs, *EAGCN* learns the invariant features from different graphs and keeps the edge attention consistency. In various experiments, we provide the in-depth analyses about property predictions, molecular graph representations and atom subtype analysis. These analyses help us understand the performance of the proposed *EAGCN* and interpret the learned models. In the future, we plan to apply *EAGCN* to other scientific domains such as social networks to evaluate its performance.

CRediT authorship contribution statement

Chao Shang: Conceptualization, Methodology, Software, Writing - original draft. **Qinqing Liu:** Data curation, Software, Validation, Visualization. **Qianqian Tong:** Methodology, Writing - review & editing. **Jiangwen Sun:** Supervision, Writing - review & editing. **Minghu Song:** Supervision, Methodology, Validation, Visualization, Writing - review & editing. **Jinbo Bi:** Supervision, Methodology, Writing - review & editing, Project administration, Funding acquisition.

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.

Acknowledgment

The work is partially supported by the National Science Foundation (NSF) under Grant No.: IIS-1320586, DBI-1356655, and IIS-1514357 to J Bi. J Bi is also supported by the NSF under Grant No.: IIS-1447711, IIS-1718738, and IIS-1407205, and the National Institutes of Health under Grant No.: K02-DA043063 and R01-DA037349.

Appendix A

The following is the mathematical proof that "Laplacian matrix L of consistent edge attention matrix is a positive semi-definite matrix".

Proof [36]. mentions that L is a real symmetric positive semi-definite matrix. However, there isn't theoretical proof for this claim. We provide the detailed proof here.

Let L_e be the Laplacian of graph G on n vertices consisting of just the edge attribute e. By additivity, the graph Laplacian $L = \sum_{e \in E} L_e$, each L_e may have different values by varying weight defined in our dictionary. Without loss of generality, we study two vertices v_1, v_2 in the weighted graph, let e be the edge of (v_1, v_2). Define the corresponding adjacency matrix as $\begin{bmatrix} w_{11} & w_{12} \\ w_{12} & w_{22} \end{bmatrix}$, degree matrix $D = \begin{bmatrix} w_{11}+w_{12} & 0 \\ 0 & w_{12}+w_{22} \end{bmatrix}$, we have:

$$D^{-1/2}AD^{-1/2} = \begin{bmatrix} \frac{w_{11}}{w_{11}+w_{12}} & \frac{w_{12}}{\sqrt{w_{12}+w_{22}}\sqrt{w_{11}+w_{12}}} \\ \frac{w_{12}}{\sqrt{w_{12}+w_{22}}\sqrt{w_{11}+w_{12}}} & \frac{w_{22}}{w_{12}+w_{22}} \end{bmatrix}.$$

The Laplacian matrix on edge e will be:

$$L_e = I - D^{-1/2}AD^{-1/2} = \begin{bmatrix} \frac{w_{11}}{w_{11}+w_{12}} & \frac{w_{12}}{\sqrt{w_{12}+w_{22}}\sqrt{w_{11}+w_{12}}} \\ \frac{w_{12}}{\sqrt{w_{12}+w_{22}}\sqrt{w_{11}+w_{12}}} & \frac{w_{22}}{w_{12}+w_{22}} \end{bmatrix}.$$

Then for any vector $x \in \mathbb{R}^n$, we get:

$$x^T L_e x = \left(\frac{\sqrt{w_{12}}}{\sqrt{w_{11}+w_{12}}} x_1 - \frac{\sqrt{w_{12}}}{\sqrt{w_{12}+w_{22}}} x_2 \right)^2 \geqslant 0.$$

If $x^T L_e x \geqslant 0$ holds for all non-zero x in R^n, the matrix is said to be a positive semi-definite matrix. It follows immediately that the Laplacian of the whole graph is positive semi-definite. $x^T L x = x^T \left(\sum_{e \in E} L_e\right) x = \sum_{e \in E} x^T L_e x \geqslant 0$, which implies that the symmetric real matrix L is a positive semi-definite matrix [37]. In addition, we also know the eigenvalues of L satisfy $0 = \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_n = \lambda_{max} \leqslant 2$ [37].

References

[1] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[2] Y. Goldberg, A primer on neural network models for natural language processing, J. Artif. Intell. Res. 57 (2016) 345–420.
[3] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.
[4] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[5] H. Liu, W. Ma, Y. Yang, J. Carbonell, Learning concept graphs from online educational data, J. Artif. Intell. Res. 55 (2016) 1059–1090.
[6] R.A. Rossi, L.K. McDowell, D.W. Aha, J. Neville, Transforming graph data for statistical relational learning, J. Artif. Intell. Res. 45 (2012) 363–441.
[7] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, IEEE Signal Process. Mag. 34 (4) (2017) 18–42.
[8] R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Machine learning: An artificial intelligence approach, Springer Science & Business Media, 2013.
[9] B. Wu, Y. Liu, B. Lang, L. Huang, Dgcnn: Disordered graph convolutional neural network based on the gaussian mixture model, Neurocomputing 321 (2018) 346–356.
[10] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M. M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model cnns, in: Proc. CVPR, Vol. 1, 2017, p. 3.
[11] J. Bruna, X. Li, Community detection with graph neural networks, arXiv preprint arXiv:1705.08415.
[12] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1025–1035.
[13] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907.
[14] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016.
[15] A. Lusci, G. Pollastri, P. Baldi, Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules, J. Chem. Inform. Model. 53 (7) (2013) 1563–1575.
[16] A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using graph convolutional networks, in: Advances in Neural Information Processing Systems, 2017, pp. 6533–6542.
[17] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, et al., Analyzing learned molecular representations for property prediction, J. Chem. Inform. Model. 59 (8) (2019) 3370–3388.
[18] N.Q.K. Le, Q.-T. Ho, E.K.Y. Yapp, Y.-Y. Ou, H.-Y. Yeh, Deepetc: A deep convolutional neural network architecture for investigating and classifying electron transport chain's complexes, Neurocomputing 375 (2020) 71–79.
[19] R. Gómez-Bombarelli, J.N. Wei, D. Duvenaud, J.M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T.D. Hirzel, R.P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS Central Sci. 4 (2) (2018) 268–276.
[20] R.C. Glen, A. Bender, C.H. Arnby, L. Carlsson, S. Boyer, J. Smith, Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to adme, IDrugs 9 (3) (2006) 199.
[21] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R. P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Advances in neural information processing systems, 2015, pp. 2224–2232.
[22] 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 (2) (2018) 513–530.
[23] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, arXiv preprint arXiv:1704.01212.
[24] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Computer-aided Molecular Design 30 (8) (2016) 595–608.
[25] D.R. Lide, CRC handbook of chemistry and physics: a ready-reference book of chemical and physical data, CRC Press, 1995.
[26] C. Chieh, Bond lengths and energies, University of Waterloo. http://www.science.uwaterloo. ca/ cchieh/cact/c120/bondel. html.
[27] A.G. Orpen, L. Brammer, F.H. Allen, O. Kennard, D.G. Watson, R. Taylor, Supplement. tables of bond lengths determined by x-ray and neutron diffraction. part 2. organometallic compounds and co-ordination complexes of the d-and f-block metals, J. Chem. Soc., Dalton Trans 12 (1989) S1–S83.
[28] M.L. Huggins, Bond energies and polarities1, J. Am. Chem. Soc. 75 (17) (1953) 4123–4126.
[29] C.W. Coley, R. Barzilay, W.H. Green, T.S. Jaakkola, K.F. Jensen, Convolutional embedding of attributed molecular graphs for physical property prediction, J. Chem. Inform. Model. 57 (8) (2017) 1757–1772.
[30] R. Khasanova, P. Frossard, Graph-based isometry invariant representation learning, arXiv preprint arXiv:1703.00356.
[31] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, J. Leskovec, Hierarchical graph representation learning with differentiable pooling, in: Advances in Neural Information Processing Systems, 2018.
[32] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Networks Learn. Syst.
[33] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in, in: International conference on machine learning, 2016, pp. 2014–2023.
[34] N. Khan, U. Chaudhuri, B. Banerjee, S. Chaudhuri, Graph convolutional network for multi-label vhr remote sensing scene recognition, Neurocomputing 357 (2019) 36–46.
[35] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, arXiv preprint arXiv:1312.6203.
[36] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[37] F. R. Chung, F. C. Graham, Spectral graph theory, no. 92, American Mathematical Soc., 1997.
[38] S. Fu, W. Liu, Y. Zhou, L. Nie, Hplapgcn: Hypergraph p-laplacian graph convolutional networks, Neurocomputing 362 (2019) 166–174.
[39] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, arXiv preprint arXiv:1506.05163.
[40] R. Levie, F. Monti, X. Bresson, M. M. Bronstein, Cayleynets: Graph convolutional neural networks with complex rational spectral filters, arXiv preprint arXiv:1705.07664.
[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, arXiv preprint arXiv:1710.10903.
[42] J. Atwood, D. Towsley, Diffusion-convolutional neural networks, in: Advances in Neural Information Processing Systems, 2016.
[43] M. Simonovsky, N. Komodakis, Dynamic edge-conditioned filters in convolutional neural networks on graphs.
[44] J.B. Lee, R.A. Rossi, S. Kim, N.K. Ahmed, E. Koh, Attention models in graphs: A survey, ACM Trans. Knowl. Discovery Data (TKDD) 13 (6) (2019) 1–25.
[45] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, D.-Y. Yeung, Gaan: Gated attention networks for learning on large and spatiotemporal graphs, arXiv preprint arXiv:1803.07294.
[46] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, A.A. Alemi, Watch your step: Learning node embeddings via graph attention, Adv. Neural Inform. Process. Syst. 31 (2018) 9180–9190.
[47] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 922–929.
[48] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, et al., Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism, Journal of Medicinal Chemistry.
[49] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, J. Huang, Self-supervised graph transformer on large-scale molecular data, Advances in Neural Information Processing Systems.
[50] R. Li, S. Wang, F. Zhu, J. Huang, Adaptive graph convolutional neural networks, arXiv preprint arXiv:1801.03226.
[51] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inform. Model. 50 (5) (2010) 742–754.
[52] A. Susnjara, N. Perraudin, D. Kressner, P. Vandergheynst, Accelerated filtering on graphs using lanczos method, arXiv preprint arXiv:1509.04537.
[53] H. Altae-Tran, B. Ramsundar, A.S. Pappu, V. Pande, Low data drug discovery with one-shot learning, ACS Central Sci. 3 (4) (2017) 283–293.
[54] D.L. Mobley, J.P. Guthrie, Freesolv: a database of experimental and calculated hydration free energies, with input files, J. Computer-Aided Mol. Des. 28 (7) (2014) 711–720.
[55] T. Niwa, B.-W. Ying, K. Saito, W. Jin, S. Takada, T. Ueda, H. Taguchi, Bimodal protein solubility distribution revealed by an aggregation analysis of the entire

ensemble of escherichia coli proteins, Proc. Nat. Acad. Sci. 106 (11) (2009) 4201–4206.
[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[57] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.
[58] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

Chao Shang received his Ph.D. at the Computer Science and Engineering Department of the University of Connecticut in 2020. He got his M.S. degree at Beijing University of Posts and Telecommunications (BUPT) in 2015. His primary research interests are in deep graph learning and machine learning using large-scale graph-structured data, with an emphasis on knowledge graphs, social networks, biomedical informatics and drug discovery.

Qinqing Liu is a Ph.D. Candidate at the Computer Science and Engineering Department of the University of Connecticut. He received his B.S. in Informatics and Computing Science from Beijing Institute of Technology, Beijing, China in 2015 and M.S. in Biostatistics from University of Connecticut, Storrs, CT, USA in 2017. His current research interests include machine learning, deep learning on graphs and 3D objects.

Qianqian Tong is a Ph.D. candidate at the Computer Science and Engineering Department of the University of Connecticut. She received her M.S. degree and B.S. in Mathematics at Zhengzhou University (China) in 2016 and 2013. Her primary Ph.D. research focuses on mathematical optimization and machine learning.

Jiangwen Sun, an assistant professor in the Department of Computer Science at Old Dominion University, received his doctorate degree in Computer Science and Engineering from University of Connecticut, master's degree in Computer Engineering from Nanjing University and bachelor's degree in Medicine from Second Military Medical University. He has broad multidisciplinary research interests, including machine learning, data mining, medical informatics and bioinformatics. He is particularly interested in pursuing precision medicine by developing and applying advanced machine learning techniques and other data analytics to analyze data from various dimensions of life at systems-level.

Minghu Song received his Ph.D. in Organic Chemistry from Rensselaer Polytechnic Institute in 2004 and M.S. in Data Science from U.C. Berkeley in 2017. Dr. Song has been working in the biotech and pharmaceutical industry for more than 15 years. He currently holds the research associate professor position in the Biomedical Engineering department at the University of Connecticut. His research focuses on developing novel computational and chemoinformatics approaches to accelerate AI-drug discovery research: 1) Develop knowledge-augmented machine learning approaches to facilitate the design-make-test-analysis cycle in drug discovery 2) Build novel knowledge graph databases to facilitate data-to-knowledge transformation and enhance data-driven statistical models 3) interpretable machine learning in drug discovery.

Jinbo Bi received her Ph.D. in Mathematics and M.Sc. in Computer Science from Rensselaer Polytechnic Institute in 2003. She is currently Frederick H. Leonhardt Chair Professor of Computer Science and Engineering at the University of Connecticut. Prior to her current appointment, she worked with Siemens Health on computer aided diagnosis research from 2003 to 2009 and Massachusetts General Hospital on clinical decision support systems from 2009 to 2010. Her research interests include machine learning, artificial intelligence, computer vision, bioinformatics, biomedical informatics, and drug discovery.