Edge Attention-based Multi-Relational Graph Convolutional Networks

Chao Shang*, Qinqing Liu*, Ko-Shin Chen*, Jiangwen Sun*, Jin Lu*

Jinfeng Yi†, Jinbo Bi*

*University of Connecticut, Storrs, CT, USA
†Tencent AI Lab, Bellevue, WA, USA
{chao.shang,qinqing.liu,ko-shin.chen,jiangwen.sun,jin.lu}@uconn.edu
jinfengyi.ustc@gmail.com,jinbo.bi@uconn.edu

Abstract

Graph convolutional network (GCN) is generalization of convolutional neural network (CNN) to work with arbitrarily structured graphs. A binary adjacency matrix is commonly used in training a GCN. Recently, the attention mechanism allows the network to learn a dynamic and adaptive aggregation of the neighborhood. We propose a new GCN model on the graphs where edges are characterized in multiple views or precisely in terms of multiple relationships. For instance, in chemical graph theory, compound structures are often represented by the hydrogen-depleted molecular graph where nodes correspond to atoms and edges correspond to chemical bonds. Multiple attributes can be important to characterize chemical bonds, such as atom pair (the types of atoms that a bond connects), aromaticity, and whether a bond is in a ring. The different attributes lead to different graph representations for the same molecule. There is growing interests in both chemistry and machine learning fields to directly learn molecular properties of compounds from the molecular graph, instead of from fingerprints predefined by chemists. The proposed GCN model, which we call edge attention-based multi-relational GCN (EAGCN), jointly learns attention weights and node features in graph convolution. For each bond attribute, a real-valued attention matrix is used to replace the binary adjacency matrix. By designing a dictionary for the edge attention, and forming the attention matrix of each molecule by looking up the dictionary, the EAGCN exploits correspondence between bonds in different molecules. The prediction of compound properties is based on the aggregated node features, which is independent of the varying molecule (graph) size. We demonstrate the efficacy of the EAGCN on multiple chemical datasets: *Tox21*, *HIV*, *Freesolv*, and *Lipophilicity*, and interpret the resultant attention weights.

Keywords Graph Convolutional Networks, Edge Attention, Graph Representation, Node Embedding

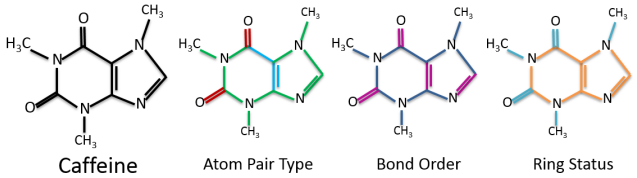
1 Introduction

Convolutional Neural Networks (CNNs) [24] have been successfully applied to study data with a grid-like structure, e.g., image, video, and speech. Such architecture offers an efficient way to extract local stationary structures and features that are shared across the data domain, and then composes them to form hierarchical patterns [23]. However, a broad range of scientific problems generate data that naturally lie in irregular spaces or generally non-Euclidean domains. There are many such examples in computational chemistry, social studies, and telecommunication networks. Data in these areas can usually be structured as graphs that encode complex geometric structures with node and edge attributes.

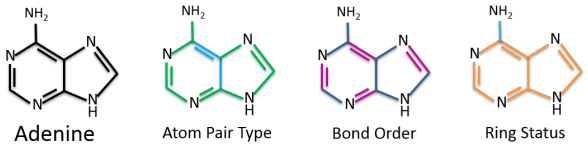
The generalization of CNNs to graph inputs is not straightforward. Due to the lack of global parameterization, common system of coordinates, vector space structure, or shift-invariance properties [3], the classical convolution operations which use fixed filter size and stride distance cannot be applied directly to graph inputs that have arbitrary structures.

Recently Graph Convolutional Networks (GCNs) have been developed and successfully tackled tasks [29] such as matrix completion [36], manifolds analysis [31], predictions on user interest/connectivity in social network [4], extracting element representations [16, 17, 22], or generating fingerprints from molecular graphs [15]. These methods are often categorized as spectral [5, 6, 9, 22] and spatial [10, 32] approaches. In spectral GCNs, node attributes are viewed as graph signals. The analysis is then performed in the spectral domain under *Graph Signal Processing* (GSP). In this framework, the convolution operation is defined through Fourier transform which is derived by graph Laplacian and its eigenspace. On the other hand, for the spatial GCNs, interactions and information exchange between two adjacent nodes are treated in a flavor of classical CNNs. By adding self-loop in the adjacency matrix, a matrix multiplication to a stack of node attributes in an entire graph automatically generates additive aggregation of a node itself and all of its neighbors.

There is a growing interest in incorporating deep learning approaches into chemoinformatics studies [11, 14, 27–29],



(a) Molecule Caffeine graph is separated into several edge-colored graphs. Only three relational graphs are given as examples, which correspond to different edge attributes. Edges in each edge relational graph are marked with different colors based on the discrete values of edge attribute.



(b) Molecule Adenine Graph and its Edge-colored Graphs.

Figure 1. Molecular graph is represented by several molecular edge-colored graphs based on the edge attributes, to represent multi-relational interactions between nodes. The attention weights we learn in our model for a specific edge attribute are shared over all molecules where the edge type is present, which enables us to extract the local stationarity properties or repeated patterns.

such as Quantitative Structure Activity Relationships (QSAR) prediction, library diversity analysis, and emerging molecular representation. Common ways to represent chemical structures are chemical formula, SMILES strings, 3D ball-and-stick, and space-fill models. Most QSAR studies on molecules have been carried out based upon some predefined molecular descriptors or fingerprints, e.g., molecular circular fingerprint [13]. In this work, we present an Edge Attention-based Multi-relational GCN (EAGCN) to predict physical chemical properties of compounds directly from the molecular graph. The EAGCN model automatically learns a set of descriptors for a molecule by dynamically aggregating the node features in graph convolution. These descriptors can be regarded as new fingerprints that may provide an effective alternative to traditional manually-defined fingerprints. Given the new fingerprints are learned during the supervised training to predict the target property, they may be more relevant to the property.

In chemical graph theory, a compound structure is often expressed as a hydrogen-depleted molecular graph whose nodes correspond to atoms in the compound while edges represent chemical bonds. Edge attributes [7] in the bond are shown in Table 1. The edge attributes are important to describe the bonding strength between two atoms, aromaticity, or bonding resonance. Each attribute may vary the

Table 1. Edge Attributes

Attribute	Description
`Atom Pair Type`	Defined by the type of the atoms that a bond connects
`Bond Order`	Bond order (bond order values have 1, 1.5, 2 and 3)
`Aromaticity`	Is aromatic
`Conjugation`	Is conjugated
`Ring Status`	Is in a ring

connectivity in a molecular graph from another attribute. If we create an attention matrix as an adjacency matrix in the graph convolution, the different edge attributes correspond to different edge attention matrices. In other words, we allow different attention weights to be learned at different layers and for different edge attributes. For instance, the edge attribute of whether a bond is in a ring assigns one of the two possible values to a bond: 1 in a ring; and 0 not in a ring. There hence will be two attention weights each corresponding to one value in our GCN model. The two weights can be used across all molecules for the bonds with this attribute. This is different from the neighborhood attention in [37]. We develop an Edge Attention Layer to estimate the weights for each edge in a molecule. We build a dictionary beforehand that consists of all possible attention weights for an edge attribute in a dataset. Then a molecule's attention matrix is constructed by looking up the dictionary for each individual bond in the molecule. Such mechanism allows different molecules to have some correspondence according to the edge attribute.

In Figure 1 for example, there are three edge attributes: Atom Pair Type, Bond Order and Ring Status, each representing a particular relation between atoms in a compound. For the edge attribute of Atom Pair Type, edges are classified according to the two atoms that an edge connects, such as C-N (green for Carbon-Nitrogen), C-O (red for Carbon-Oxygen), or C-C (blue for Carbon-Carbon) etc. For the edge attribute of Bond Order, edges are classified as single bond (dark blue), and double bond (purple). For the ring status, edges are classified by whether it is in a ring (orange) or not (cyan). These colors (actually weights) are unique for all compounds in a single dataset.

Since edge attentions are shared across all graphs, our EAGCN method also extracts invariant properties of graphs [20]. In graph theory, graph invariant is defined as a property preserved under all possible isomorphisms of a graph, which includes the order invariant, permutation invariant and pair order invariant [19]. In Figure 1, we can see that our model will learn the edge attention that is shared at intra-molecule and inter-molecule levels, which means that edges are in a sense aligned using the same edge attention. In addition, our

model can also handle the varying graph sizes because of the edge attribute based edge alignment which is in contrast to most previous works that were focused on a big graph or fix-sized unaligned graphs.

Our major contributions are summarized as follows:

1. We propose an Edge Attention based Multi-relational Graph Convolutional Networks, an efficient model to learn multiple relational strengths of node interaction from neighbors.
2. Aligning attention weights for the same type of edge universally across all molecules. In other words, the EAGCN learns invariant features from inherent invariant properties of the graphs.
3. The input graphs of our model can have varying sizes.
4. Edge attention weights in the attention adjacency matrices give some insights on how atoms influence each other and how this influence relates to the target property.

Empirical evaluation of the proposed approach on four real-world datasets demonstrates the superior performance of the EAGCN on molecular property prediction. The rest of the paper will proceed as follows. In Section 2 we discuss related works. Section 3 is dedicated to the description of our method followed by a summary of experimental results in Section 4. We then conclude in Section 5 with a discussion of future works. Our source code is publicly available at https://github.com/Luckick/EAGCN.

2 Related Work

2.1 Spectral Graph Convolutions

The graph convolutional networks were first proposed in [5] where graph convolutional operations were defined in the Fourier domain. Since the eigendecomposition of the graph Laplacian is needed, it involves intense computations. Later, smooth parametric spectral filters [18] were introduced to achieve localization in the spatial domain and the computational efficiency. In particular, Chebyshev polynomials [9] and Cayley polynomials [25] have been utilized in these convolutional architectures to efficiently produce localized filters. Recently, Kipf et al. [22] simplified these spectral methods by a first-order approximation of the Chebyshev polynomials. Their derivation finally leads to a one-step neighbors localization and achieves state-of-the-art performance.

However, the spectral filters learned by 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 a set of graphs with different structures.

Recently graph attention networks [37] have been proposed to deal with arbitrarily graphs without knowing the entire graph structures. The attention mechanisms allow the model to deal with varying size inputs. This attention-based architecture assigns different importances to different nodes within a neighborhood while dealing with various sized neighborhoods. However, it is node-similarity driven and has not considered the edge information that could raise different attentions. Inspired by this work, we present an edge attention mechanism that identifies edge importances for the property prediction. Unlike existing approaches, we leverage edge attributes to predict multiple relational strengths of connection and interaction between nodes for better graph representation and property prediction.

2.2 Non-spectral Graph Convolutions

The spatial graph convolution approaches [2, 10, 17] define convolutions directly on graph, which sum up node features over all spatially close neighbors by using adjacency matrix. The main challenge is from the dynamically sized neighborhoods that bring a difficulty to maintain the weights sharing property. Duvenaud [10] presented a convolutional neural network that operates directly on raw molecular graphs. It learns a specific weighted matrix for each node degree. The approach in [32] selected fixed-size neighbors and normalized these nodes, which enabled the traditional CNNs to be applied to graph inputs directly. In order to handle graphs of varying size and connectivity, Simonovsky et al. [34] proposed the edge convolution network (ECC) for point cloud classification. Their model defines several node feature filters based on the edge label. Recently Hamilton et al. [17] introduced the task of inductive node classification, where the goal is to classify nodes that were not seen during training. This approach samples a fixed-size neighbors from each node and achieves state-of-the-art performance across several datasets.

2.3 Convolutions on Molecular Graph

In chemistry field, neural networks and GCNs have been applied to studies such as protein interface prediction[11], molecular representation and prediction [7, 10, 12, 19, 26]. The work [10] presented a convolutional neural network that operates directly on raw molecular graphs and generalizes standard molecular feature extraction methods based on circular fingerprints (ECFP) [33]. Based on Autoencoder model, [15] converted discrete representations of molecules to a multidimensional continuous one. To gain additional information from bonds, the following methods have been proposed. Here the bonds are labeled with numerous attributes including the atom-pair type or the bond order. [19] supported graph-based model that utilizes properties of both nodes (atoms) and edges (bonds). [7] created the atom feature vectors concatenated with their respective connecting bonds' features to form atom-bond feature vectors. In these works, node features and bond attributes are treated equally instead of internal relation. However, as we pointed out before, edge attentions imply various interaction types between atomic pairs. Thus they shouldn't be treated equally during node

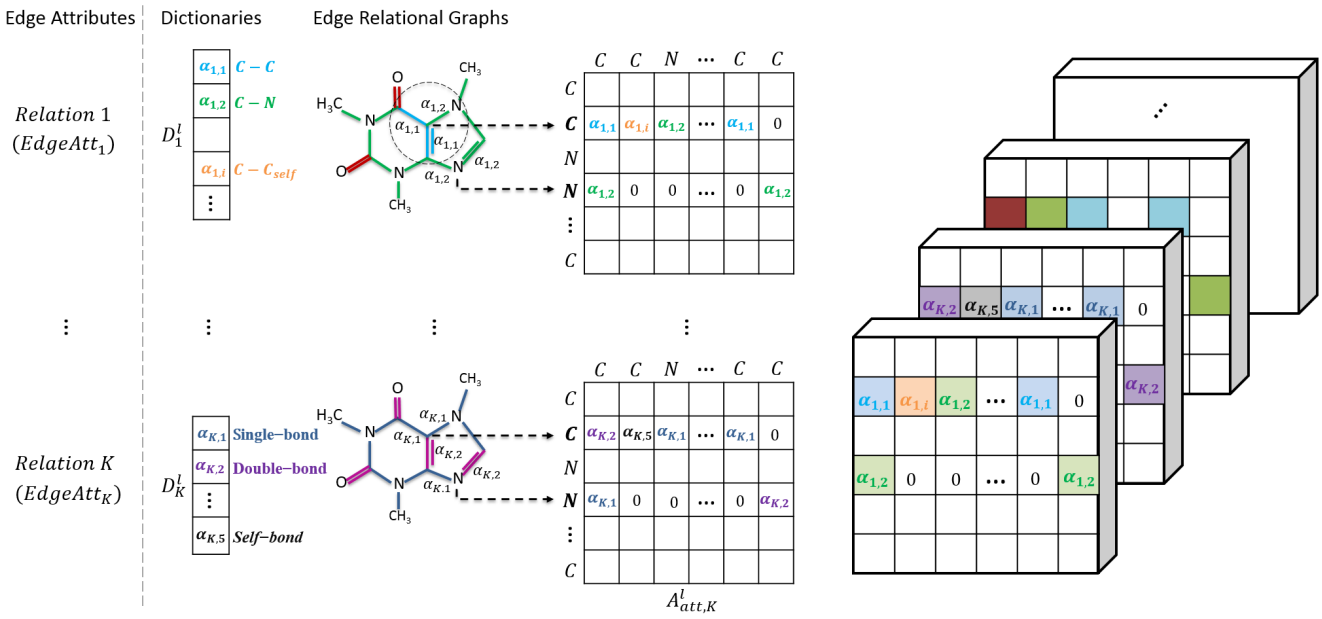


Figure 2. The process to build the weighted adjacency tensor.

feature aggregation.

3 Method

In this section, we give a detailed description of edge attention based multi-relational GCN (EAGCN) model. We aggregate node feature information of each node in a graph with all of its neighbors based on node-to-node interactions. Here we assume an interaction between two adjacent nodes is governed by edge attributes. As edge attributes are considered to have discrete values, different types of a single edge attribute contribute different strengths of interactions. This allows our model to learn multiple relationships of paired nodes across all edge attributes.

3.1 Edge Attention Layer

In EAGCN, each attention layer learns a set of adjacent matrices such that each weight corresponds to a strength of interaction from a particular edge type. We first introduce the following notations used in our formulations.

- A graph is denoted by $G = (V, E)$, where V is a finite set of nodes with $|V| = N$, and $E \subseteq V \times V$ is a finite set of edges with $|E| = M$.
- An adjacency matrix A of G is a square binary matrix. An element $a_{ij} = 1$ indicates that there is an edge between nodes i and j.
- For the layer of index l, the input contains a node feature matrix $H^l \in \mathbb{R}^N \times \mathbb{R}^F$, where the i-th row represents features of the node i and a set of edge attributes $\{\vec{f}(e) \in \mathcal{S}_1 \times \cdots \times \mathcal{S}_K | e \in E\}$. Here $\{\mathcal{S}_j\}$ are discrete sets, F is the number of features in each node, and K is the number of edge attributes.
- We assume, for edge attribute i, there are d_i possible outcomes (types) i.e. $|S_i| = d_i$.
- The linear transformation from the input of the layer l to its output is parametrized by matrix coefficients $\{W_i^l \in \mathbb{R}^F \times \mathbb{R}^{F_i'} | 1 \le i \le K\}$.
- The output of the layer l is a set of (aggregated) node feature matrices: $\mathbf{H}^{l+1} = \{H_i^{l+1} \in \mathbb{R}^N \times \mathbb{R}^{F_i'} | 1 \le i \le K\}$.

As shown in Figure 2, for the edge attribute $EdgeAtt_i$, the outcome contains d_i different types. To assign the weight of interaction for each connection type, we build a dictionary $\mathbf{D}_i^l \in \mathbb{R}^{d_i}$ which will be learned by our EAGCN model. Here we emphasize the dictionary is a fixed structure for one dataset and each molecule is coded based on its specific edge attributes. In addition, the dictionary for each edge attribute is not only shared for one graph, but also used for all graphs in the dataset. Then a weighted adjacency matrix $A_{att,i}$ corresponding to $EdgeAtt_i$ is constructed according to this dictionary. Here an element $\alpha_{i,j}$ in the matrix , coming from the dictionary, denotes the weight of the edge type j in $EdgeAtt_i$.

Based on dictionaries obtained from K different edge attributes, we can draw multiple types of edge-colored edge relational graphs. In the study of chemical compounds for example, such collection of graphs gives different perspectives of atomic interaction and strength of influence. We then call the stack $[A_{att,1}, A_{att,2}, ..., A_{att,K}]$ *edge attention* weighted adjacency tensor.

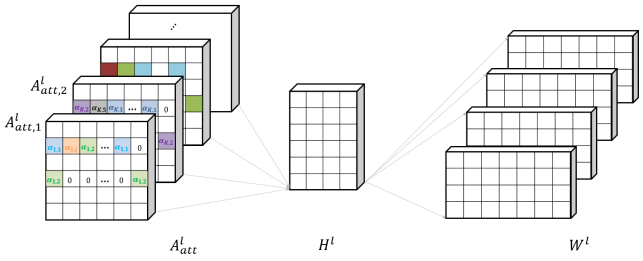


Figure 3. Edge Attention Convolution.

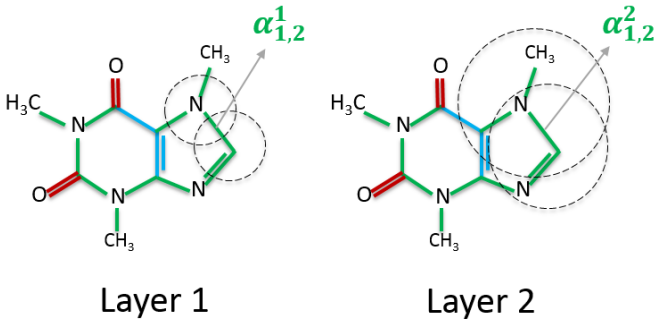


Figure 4. Learn the attention from different scopes in different layers. Each scope has different substructure information.

Architecture of $A_{att,i}$

In layer l, we first create a one-hot encoding for the edge attribute i. This yields a binary vector $\mathbf{mask}_i(e) \in \mathbb{R}^{d_i}$ for each edge $e \in E$. Furthermore, the adjacency matrix A turns to be a one-hot encoding adjacency tensor $T_i \in \mathbb{R}^{N\times N\times d_i}$ such that if the element $(A)_{s,t} = 0$ i.e. there is no edge between nodes s and t, $(T_i)_{s,t}$ will be a zero vector. Finally, we obtain the weight $\alpha^l_{i,j(e)}$ in $A^l_{att,i}$ by

$$\alpha^l_{i,j(e)} = \langle \mathbf{mask}_i(e), \mathbf{D}^l_i \rangle \tag{1}$$

Note that we may view the process of $T_i \longrightarrow A^l_{att,i}$ as a special case of 'image' convolution with d_i input channels and one output channel. The filter $\mathbf{D}^l_i$ with size $1 \times 1 \times d_i$ is moving with stride of 1.

In our experiments, in order to make coefficients comparable cross different edges, we normalize the weights using the softmax function:

$$(\tilde{A}_{att,i})_{s,t} = \frac{exp(A^l_{att,i})_{s,t}}{\sum_{t=1}^{M} exp(A^l_{att,i})_{s,t}} \tag{2}$$

Edge Attention Convolution

In each graph convolution layer, we consider the node information aggregation over all first-order neighbors followed by a linear transformation:

$$H^{l+1}_i = \sigma(\tilde{A}_{att,i} H^l W^l_i) \tag{3}$$

for $1 \le i \le K$, where σ is an activation function. As shown in Figure 3, each edge attribute i generates a single aspect of interaction $\tilde{A}_{att,i}$ and the term $\tilde{A}_{att,i}H^l$ can be viewed as a weighted sum of node features. After computing $\mathbf{H}^{l+1} = \{H^{l+1}_i \in \mathbb{R}^N \times \mathbb{R}^{F'_i} | 1 \le i \le K\}$, we define a function $P : \Pi^K_{i=1}(\mathbb{R}^N \times \mathbb{R}^{F'_i}) \longrightarrow \mathbb{R}^N \times \mathbb{R}^C$ such that the output $H^{l+1} = P(\mathbf{H}^{l+1})$.

In this work, we implement our model using two different settings on the function P. One way is to concatenate all $H^{l+1}_i s'$ to form a matrix of dimension $\mathbb{R}^N \times \mathbb{R}^{F'_1+F'_2+\dots F'_K}$:

$$H^{l+1} = P_{concat}(\mathbf{H}^{l+1}) := (H^{l+1}_1, H^{l+1}_2, ..., H^{l+1}_K). \tag{4}$$

Another way is the weighted sum which is applicable only when $F' := F'_1 = F'_2 = ... = F'_K$:

$$H^{l+1} = P_{w-sum}(\mathbf{H}^{l+1}) := \sum_{i=1}^{K} \beta_i H^{l+1}_i. \tag{5}$$

The experiment results for both settings will be given in Section 4.

EAGCN Framework

From (3), we see that for each node in a graph, 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 from higher layers learn the interactions of substructures. In Figure 4 for example, the $\alpha^1_{1,2}$ in Layer 1 represents the atomic interaction between Nitrogen and Carbon; the $\alpha^2_{1,2}$ in Layer 2 represents the interaction between two bond groups centered at Nitrogen and Carbon since the information from neighbors is already gathered from the second order neighbors i.e. neighbors' neighbors. Thus the attention weights are capable of characterizing substructure within different scopes.

3.2 Graph invariance and varying graph size

Firstly, for each edge attribute, we have created a dictionary with learnable weights, which implies the strengths of connection and interaction between nodes. The attention weights are conditioned on this dictionary in the neighborhood of a node, instead of the neighborhood order. This dictionary results in a homogeneous view for local graph neighborhoods. These weights are not only shared in one graph, but also for all graphs, which enable us to extract the local stationarity property of the input data by revealing local features what are shared across all graphs. Hence, EAGCN model has been designed to produce invariant features by replacing neighbors' attention to the edge attention to deal with graph invariance problem [19].

Secondly, varying graph size problem can also be solved by the attention mechanism. For each edge attention layer, the number of parameters in each attention matrix is the number of edge attribute weights occurred in each dictionary instead of number of edges. Therefore, EAGCN model is insensitive to the varying graph sizes.

4 Experiments

4.1 Benchmark Datasets

Four benchmark datasets[1, 38] (Tox21, HIV, Freesolv and Lipophilicity) are utilized in this study to evaluate the predictive performance of built graph convolutional networks. They are all downloaded from the MoleculeNet website[1] that hold various benchmark datasets for molecular machine learning.

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 that measure various nuclear receptor or oxidative stress responses, e.g., androgen receptor, estrogen receptor, and mitochondrial membrane potential.

HIV The HIV dataset was introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen, which tested the ability to inhibit HIV replication for over 41,127 compounds. Screening results were evaluated and placed into three categories: confirmed inactive (CI), confirmed active (CA) and confirmed moderately active (CM). We combine the latter two labels for a classification task.

Freesolv Freesolv is a database of experimental and calculated hydration free energies for small neutral molecules in water, along with molecular structures, input files, references, and annotations [30]. It includes a set of 642 neutral molecules which are mostly fragment-like. The calculated values are derived from alchemical free energy calculations using molecular dynamics simulations.

Lipophilicity (Lipo) Lipophilicity, curated from ChEMBL database, provides experimental results of octanol/water distribution coefficient of 4200 compounds. Lipophilicity is an important feature of drug molecules that affects both membrane permeability and solubility, which is used for the regression task.

4.2 Experimental Setup

Our experiments evaluate the property prediction on standard supervised classification and regression tasks. We design our experiments with the goals of 1) verifying the improvement of our method compared with baseline methods, such as GCN [22] and 2) comparing two different architectures of our method. We adapt two edge attention layers and three fully connected layers for graph classification and regression.

The node features and edge attributes are extracted using the RDKit[2], 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 graph. Here we ignore the SMILES samples whose structure graphs have no edge. The edge attributes [7] are shown in the Table 1. When we build the dictionary for 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 will set one attention weight for them in the dictionary. Each data is randomly split into three sets: training (81%), validation (9%), and testing (10%). Then three independent runs with different random seeds are performed. Note that all results presented here are the average of three runs, with standard deviations listed. We use the adaptive moment (Adam) algorithm [21] for training the model and set the learning rate to 0.0005 for classification tasks and 0.001 for regression tasks. Our models[3] are implemented by PyTorch and run on Ubuntu Linux 14.04 with NVIDIA Tesla K40C Graphics Processing Units (GPUs).

4.3 Baselines

We compare our model with the five baseline methods which are shown in MoleculeNet [38]. Firstly, the Kernel-SVM [8, 35], one of the most famous machine learning method, is used for the classification task. The second method is Random Forests (RF), which can be used for both classification and regression tasks. A random forest consists of many individual decision trees. The output predictions are the average results from all trees. Thirdly, the graph convolutions network (GCN) [22] is the baseline for the comparison. We also compare with another two models which leverage the edge attributes. Weave model [19] is similar to graph convolutions, the weave featurization encodes both local chemical environment and connectivity of atoms in a molecule. The weave featurization calculates a feature vector for each pair of atoms in the molecule. Message passing neural network (MPNN) [12] is a generalized model, which have two phases. Multiple message passing phases are stacked to extract abstract information of the graph, then the readout phase is responsible for mapping the graph to its properties. We compare with these models to show that EAGCN provides a novel and efficient way to use edge attributes.

4.4 Classification Analysis

Table 2 reports ROC-AUC results of four different baseline models on biophysics datasets (HIV) and physiology dataset (Tox21). These two datasets contain only classification tasks.

[1] http://moleculenet.ai

[2] http://www.rdkit.org

[3] https://github.com/Luckick/EAGCN

Table 2. Prediction results for the four benchmark datasets. EAGCN with concatenation and EAGCN with weighted sum are shown in comparison with five baseline approaches.

Task	Classification (AUC)				Regression (RMSE)			
Dataset	Tox21		HIV		Lipo		Freesolv	
	Validation	Testing	Validation	Testing	Validation	Testing	Validation	Testing
Kernel-SVM	0.78 ± 0.01	0.77 ± 0.02	0.75 ± 0.04	0.76 ± 0.01	—	—	—	—
RF	0.78 ± 0.01	0.75 ± 0.03	**0.83 ± 0.02**	0.82 ± 0.02	0.87 ± 0.02	0.86 ± 0.04	1.98 ± 0.07	1.62 ± 0.14
Weave	0.79 ± 0.02	0.80 ± 0.02	0.68 ± 0.03	0.71 ± 0.05	0.88 ± 0.06	0.89 ± 0.04	1.35 ± 0.22	1.37 ± 0.14
MPNN	—	—	—	—	0.88 ± 0.01	0.88 ± 0.02	1.09 ± 0.25	1.15 ± 0.43
GCN	0.82 ± 0.02	0.84 ± 0.01	0.70 ± 0.05	0.77 ± 0.02	0.66 ± 0.05	0.68 ± 0.03	1.30 ± 0.09	1.35 ± 0.26
EAGCN_{w-sum}	0.85 ± 0.00	0.86 ± 0.00	0.79 ± 0.03	0.81 ± 0.01	0.64 ± 0.09	0.64 ± 0.01	**1.02 ± 0.33**	**0.92 ± 0.09**
EAGCN_{concat}	**0.85 ± 0.00**	**0.86 ± 0.01**	0.80 ± 0.03	**0.83 ± 0.01**	**0.61 ± 0.05**	**0.61 ± 0.02**	1.05 ± 0.23	0.95 ± 0.14

Three independent runs with different random seeds are performed. For our EAGCN model, we implement our model using two different settings on the function P. EAGCN_{concat} is to concatenate all feature matrices, and EAGCN_{w-sum} is the weighted sum.

Comparison of models

In the paper [38], there are several baseline models. For Tox21, GCN and Weave achieve the best performances in the test and validation datasets. In Table 2, GCN, and Weave also get the best performances in the test dataset. Our EAGCN model improves upon GCN by a margin of 2.4%, and upon Weave by a margin of 7.5% for the test. For HIV, GCN and kernel-SVM achieve the best performances for the test dataset in the [38]. In Table 2, GCN achieves the best performance in the test dataset and Kernel-SVM also gets a nice result, which is consistent with the paper. Our EAGCN model improves upon GCN by a margin of 7.8% for the test dataset. Here we also run the RF method and find that RF can get an excellent performance. We improve upon RF by a margin of 1.2% for the test dataset. Our EAGCN always achieves reasonable and excellent performance for prediction of properties, which has strong validation/test results on the datasets. In addition, our approaches don't exhibit the large gaps between train scores and validation/test scores.

The classification model building upon the Tox21 dataset can be further utilized to identify any new compounds with potential liability associated with the above 12 response pathway and prioritize specific compounds for more extensive toxicological evaluation. Figure 5 shows the performance for 12 output nodes, one for each of the prediction targets. For almost all the tasks, our two approaches and GCN achieve higher AUC scores for all the targets except "HR-AhR", where the performance of RF is a little bit better than GCN. We emphasize that this performance was achieved by learning directly from the molecular graph, rather than from precomputed properties. In summary, EAGCN_{w-sum} and

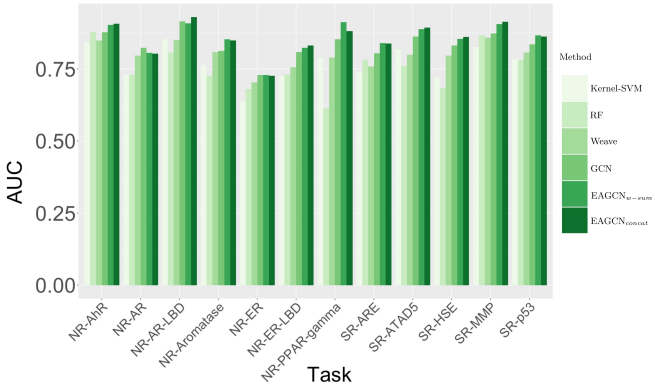


Figure 5. Model performance for these 12 targets in comparison to four baseline models.

EAGCN_{concat} outperformed other methods on eleven targets over twelve.

4.5 Regression Analysis

Solubility and lipophilicity are basic physical chemistry properties important for understanding how molecules interact with solvents. Our method in the Table 2 performs at the level of state of the art for Lipophilicity and Freesolv datasets, which contain only regression tasks.

Comparison of models

First, we compare with the best baseline model in the paper [38] for the regression tasks. In Lipophilicity dataset, the best baseline model in [38] is GCN. In the table, we get the same conclusion for GCN model. We achieve about 7.6% and 10.3% performance increases for the validation and test datasets. For the Freesolv dataset, the best baseline model in our table is same with the best model in [38]. We improve upon MPNN by a margin of 6.4% and 20.0% for the validation and test datasets.

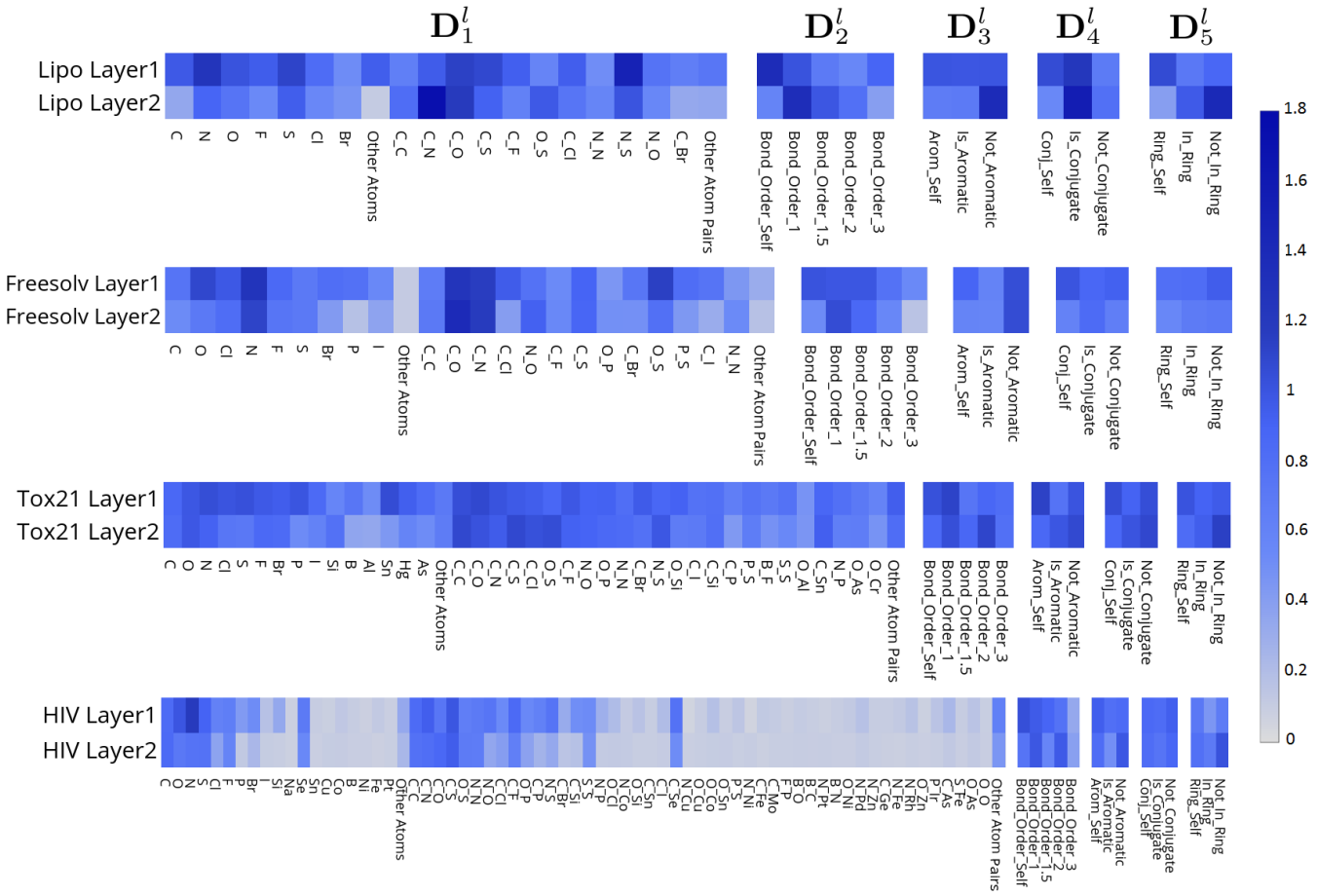


Figure 6. Visualization of edge attention dictionaries.

EAGCN shows strong validation/test results on larger dataset Lipo, illustrating that learnable featurizations can provide a large boost compared with conventional featurizations. Given the size of FreeSolv dataset is only around 600 compounds, our model can still reach excellent performances by training on limited samples. In addition, the larger dataset, GCN has a smaller standard deviation to get a more robust estimate. The best-performing model in this table is EAGCN, which can reach the small RMSE of 0.61. 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 two datasets, data-driven methods can outperform physical algorithms with moderate amounts of data. These results suggest that graph convolutional approaches will become increasingly important for the property prediction.

4.6 Visualization and Chemical Analysis

In physical chemistry, the edge attention mechanism learns multiple type strengths of influence from neighbors, which gives some insights on atomic interaction. In Figure 6, we visualize the edge attention dictionaries using four heatmaps. The darkness of a block corresponds to the attention weight value. And the darkness is consistent for all datasets. For each dataset, we have five edge attribute dictionaries $D_i^l (1 \leq i \leq 5)$ in each layer l, collectively represented the strengths of atomic interaction. In our experiments, we have two edge attention layers so that we have two rows in each dataset. The label of each column is the edge attention type. For different datasets, we have different number of edge attention types in the five dictionaries. The edge attention types are the discrete values of edge attributes. The labels of single elements are the self-attentions, which means that we add a self-loop for each atom. The reason is the multiplication with each weighted adjacency matrix for each edge attribute means, for every node, we sum up all the feature vectors of all neighboring nodes but not the node itself. We fix this by enforcing self-loops in the graph.

Firstly, we can see that many same attention types from different datasets have the similar weight values. Our explanation is that these weights learn the general atomic interactions which are unrelated with the tasks. Secondly, as is shown in the section 3.1, the different attention layers learn the interaction between paired nodes in different scopes. Hence the attention weight in different layers has the different weight values. In the Figure 6, Freesolv dictionaries have the $C-O$ and $C-N$ attention weights, which both have higher attention values in layer 1 and layer 2. For $C-O$ pair, the results mean the substructure around C has a significant strength of influence to the substructure around O for the solubility property. Thirdly, the $C-Sn$ attention value is quite different between Tox21 and HIV. The $C-Sn$ has higher values of both layers in Tox21, but its values in both layers are very low in HIV. Our explanation is that the interaction between C and Sn is associated with toxicity prediction instead of the ability to inhibit HIV replication. In addition, the dictionaries for HIV are very interesting for us. We can clearly see the difference between different attention types, indicating the different levels of importance, which gives a new guidance to exploit how to inhibit HIV replication.

5 Conclusion

We have introduced an Edge Attention based Multi-relational Graph Convolutional Network (EAGCN), which builds multiple relational connections parameterized by edge attention weights for graph representation and molecular property prediction. Our edge attentions layers estimate the edge attribute based edge attentions from different atomic pairs, which has several attractive qualities. Firstly, the edge attention mechanism allows us to learn multiple relational strengths of node interaction from neighbors. Secondly, since attention weights from dictionaries are shared across all graphs to extract the local stationarity property, EAGCN model produces invariant features for inherent invariant properties of graphs. Thirdly, our model can also handle the varying graph size because of edge alignment. Finally, the model learns an attention adjacency tensor, which gives some insights on how atoms influence each other. In the future, we will extend our model to multiple types of data in different situations.

Acknowledgments

The authors would like to thank Dr. Minghu Song from the Center for Molecular Discovery at Yale University for discussions on model construction and chemical insights. They acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40C GPU. The work is partially supported by the National Science Foundation (NSF) under Grant No.: IIS-1320586, DBI-1356655, and IIS-1514357. Jinbo 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.

References

[1] Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. 2017. Low data drug discovery with one-shot learning. *ACS central science* 3, 4 (2017), 283–293.

[2] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1993–2001.

[3] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42.

[4] Joan Bruna and Xiang Li. 2017. Community detection with graph neural networks. *arXiv preprint arXiv:1705.08415* (2017).

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[6] Fan RK Chung. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

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

[8] Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*. 3844–3852.

[10] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*. 2224–2232.

[11] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems*. 6533–6542.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212* (2017).

[13] Robert C Glen, Andreas Bender, Catrin H Arnby, Lars Carlsson, Scott Boyer, and James Smith. 2006. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to ADME. *IDrugs* 9, 3 (2006), 199.

[14] Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. 2017. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603* (2017).

[15] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. 2016. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science* (2016).

[16] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1025–1035.

[18] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).

[19] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design* 30, 8 (2016), 595–608.

[20] Renata Khasanova and Pascal Frossard. 2017. Graph-based isometry invariant representation learning. *arXiv preprint arXiv:1703.00356* (2017).

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[25] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. 2017. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *arXiv preprint arXiv:1705.07664* (2017).

[26] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive Graph Convolutional Neural Networks. *arXiv preprint arXiv:1801.03226* (2018).

[27] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. 2013. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling* 53, 7 (2013), 1563–1575.

[28] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. 2016. DeepTox: toxicity prediction using deep learning. *Frontiers in Environmental Science* 3 (2016), 80.

[29] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. 2013. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

[30] David L Mobley and J Peter Guthrie. 2014. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design* 28, 7 (2014), 711–720.

[31] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proc. CVPR*, Vol. 1. 3.

[32] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *International conference on machine learning*. 2014–2023.

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

[34] Martin Simonovsky and Nikos Komodakis. [n. d.]. Dynamic edge-conditioned filters in convolutional neural networks on graphs.

[35] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.

[36] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *stat* 1050 (2017), 7.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903* (2017).

[38] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* 9, 2 (2018), 513–530.