# Predicting Outcomes of Chemical Reactions: A Seq2Seq Approach with Multi-view Attention and Edge Embedding

Xia Xiao, Chao Shang, Jinbo Bi, Sanguthevar Rajasekaran
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA
{xia.xiao, chao.shang, jinbo.bi, sanguthevar.rajasekaran}@uconn.edu

***Abstract*—Materials Genomics initiative has the goal of rapidly synthesizing materials with a given set of desired properties using data science techniques. An important step in this direction is the ability to predict the outcomes of complex chemical reactions. Some graph-based feature learning algorithms have been proposed recently. However, the comprehensive relationship between atoms or structures is not learned properly and not explainable, and multiple graphs cannot be handled. In this paper, chemical reaction processes are formulated as translation processes. Both atoms and edges are mapped to vectors representing the structural information. We employ the graph convolution layers to learn meaningful information of atom graphs, and further employ its variations, message passing networks (MPNN) and edge attention graph convolution network (EAGCN) to learn edge representations. Particularly, multi-view EAGCN groups and maps edges to a set of representations for the properties of the chemical bond between atoms from multiple views. Each bond is viewed from its atom type, bond type, distance and neighbor environment. The final node and edge representations are mapped to a sequence defined by the SMILES of the molecule and then fed to a decoder model with attention. To make full usage of multi-view information, we propose multi-view attention model to handle self correlation inside each atom or edge, and mutual correlation between edges and atoms, both of which are important in chemical reaction processes. We have evaluated our method on the standard benchmark datasets (that have been used by all the prior works), and the results show that edge embedding with multi-view attention achieves superior accuracy compared to existing techniques.**

***Index Terms*—Computational Chemistry, Attention Model, Representation Learning.**

## I. Introduction

Synthesis of novel materials with desired properties is a central problem of enormous economic implications in materials science and Materials Genomics, in particular. In this context, the construction of a target molecule from a set of existing reactants and reagents is of interest. If a molecule can be represented as a string, the search space of possible candidate outcomes is $O(k^l)$, where $l$ is the length of the outcome, and $k$ is the number of different types of atoms appearing in the reactants. Extensive research has been conducted in the past decades to formalize the outcomes of reactions. This formalization forms the basis for solving synthesis problems.

This work has been supported in part by the following NSF grants: 1447711, 1514357, 1743418, and 1843025.

The most challenging task in modeling any chemical reaction process is to locate the reaction sites. The reaction sites are the locations that re-organize the molecular graph, or SMILES string. However, the naive SMILES with one-hot or simple encoding fails to reflect the underlying chemical or physical properties of the atoms or edges. As a consequence, a complex prediction model directly applied to one-hot encoded SMILES requires a significantly large number of trainable parameters and long training times. Proper representation is required to reduce the complexity and improve the accuracy and generalization of the model. However, the widely used simplified molecular-input line-entry system (SMILES) [1] lacks the structural and distance information.

Currently, molecular representation learning focuses on atom embedding. Graph Convolutional Networks (GCN) based methods have been developed and successfully used to address tasks [2] such as matrix completion [3], social networks analysis [4], element representations [5], [6], and generating fingerprints from molecular graphs [7]. For molecule representation learning, the model can potentially learn the structural information inside molecules. However, previous works neglect the diverse properties of chemical bonds which could play an important role in outcomes prediction. A good representation should reflect the bond energy of a chemical bond during the chemical reaction process, which directly determines the possibility of being a reaction site. [8] proposed a multi-step update rule to learn representations for both nodes and edges, which is suitable to learn edge information in molecule graphs. [9] proposed a multi-view approach where the edges were grouped such that the edges in the same group have similar properties. The edge representation is defined as a function of its atom pair type, bond order, aromaticity, conjugation, and ring status. The paper shows promising results for molecular embedding and property prediction. In this paper, we consider three methods GCN, MPNN and EAGCN in the molecule graph embedding procedure.

Given a property molecular representation, we further need to design an efficient model for molecule generation. Possible candidates are generating graphs and Long-Short Term Memory (LSTM) based sequence to sequence (Seq2Seq) translation models. Graph generating models can be embedded naturally with GCN representation learning. However, generating graphs

are much more challenging to process than generating strings due to the extra geometric information of graphs. The advantage of employing translation models compared to graph-based models is the feasibility of the generation process. In this paper, we first embed the atoms and bonds through an edge embedding graph convolution neural network, and then feed to Seq2Seq model to generate the outcomes. During the decoding process, the decoder outputs atoms by considering the corresponding chemical bonds through attention mechanisms to determine the reaction site. We evaluate the effectiveness of the learned representation by comparing the output accuracy with pure Seq2Seq and atom embedding models. We also show that the multi-view representation is more efficient than single-view representation learning.

In [10], the authors show the effectiveness of applying Seq2Seq models to predict chemical reaction outcomes. However, they do not utilize the structure of the molecules. For instance, reaction outcomes could depend on interatomic distances which are ignored in the algorithm. Moreover, traditional attention mechanics in Seq2Seq models can only handle single-view (inter-atom) correlations, which is not sufficient when multiple properties are considered. In the chemical reaction process, multiple correlation between different properties of atoms and between atom and edge should be considered. In this paper, we take two way attention to model the interaction between atom embedding view and edge embedding view(mutual correlation) and multi-way attention to handle multiple view inside each atom or edge embedding(self correlation). The attention factors from different views are finally combined and the final attention score is calculated as part of inputs to the Seq2Seq decoder.

Salient features of our work are: 1) We propose a novel multi-view attention mechanic in Seq2Seq model for molecule prediction. Both self and mutual correlations are calculated to generate final attention score. 2) Our approach significantly improves the prediction accuracy for predicting the outcomes of complex chemical reactions. 3) We compare different embedding methods and provide meaningful results. We conclude that the edge embedding can extract meaningful and distinguishing information for the molecules. 4) Our attention results provide certain extent of interpretability on importance of different sites.

## II. Related Work

The predicting models for chemical reactions can be categorized into two types: template-based models and template-free models. Due to the template coverage and the complexity of a chemical reaction, template-free method is more suitable for the problem.

[11] have shown that organic molecules contain fragments whose rank distribution is, to some extent, identical to that of sentence fragments. Their results indicate that organic chemistry and human language follow very similar laws, which provides guidance to use linguistics-based analyses in the area of chemical reactions. [12] used a novel approach based on Weisfeiler-Lehman Networks (WLN). They trained two independent networks on a set of 400,000 reactions extracted from US patents. The first WLN scored the reactivity between atom pairs and predicted the reaction center. All possible bond configuration changes were enumerated to generate product candidates. The candidates that were not removed by hard-coded valence and connectivity rules are then ranked by a Weisfeiler-Lehman Difference Network (WLDN). Jin, et al., [12] claimed to outperform template-based approaches by a margin of 10% after augmenting the model with the unknown products of the initial prediction to have a product coverage of 100% on the test set. Nam and Kim [13] used a template-free Seq2Seq model to predict reaction outcomes. Whereas their network was trained end-to-end on patent data and self-generated reaction examples, they limited their predictions to textbook reactions. Further, the authors of [10] view the reaction prediction task as a translation problem and solve it using natural language processing methods such as Sequence to Sequence (Seq2Seq) models. The model is designed to learn the mapping from the input sequence to the output sequence directly, based on the statistical relationships among the atoms, instead of using expert created and/or machine learned rules. Any molecule is represented as a sequence using the SMILES. However, both [13] and [10] failed to employ the structural information of a molecule graph.

For molecular graph embedding, GCNs have been employed as protein interface prediction [14], molecular representation and prediction [6], [15], [16]. The work [17] presented a convolutional neural network that operates directly on raw molecular graphs and generalizes standard molecular feature extraction methods based on circular fingerprints (ECFP) [18]. Based on the autoencoder model, [7] 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. [16] proposed a graph-based model that utilizes the properties of both the nodes (atoms) and the edges (bonds) by calculating an edge matrix for all pairs of atoms. Similarly, [19] created atom feature vectors concatenated with their respective connecting bond features to form atom-bond feature vectors. In these works, node features and bond attributes are treated equally. In [15], the author proposed a message passing network that aggregates the local information form the neighbor nodes. Both the node and edge representation can be learned. However, edge attentions imply various interaction types between atomic pairs. The diversities of edges are of great importance for the chemical reaction. For instance, [20] has proposed an attention framework to update the edge representation based on the structure of the graph. While this method can handle single large graphs well, it is not suitable for multi-graph datasets, since the learned attention weights from one graph cannot be applied to another graph.

## III. Methods

We present an end-to-end learning framework for the chemical reaction prediction. The framework consists of two compo-

nents: graph embedding for learning comprehensive node and edge representations, and an attention-based Seq2Seq model for generating the outcomes of chemical reactions.

Given the SMILES strings as the inputs, we first convert strings into molecular graphs using the RDKIT package. The first step is to employ two edge attention graph convolution layers (EAGCN) [9] to learn atom feature vectors and the edge feature vectors. After this step, the node edge representation is fed to a sequence to sequence model. The whole Graph2Seq model is trained with true dependency from the graphs. The attention layer and the decoder will focus on not only atoms but also chemical bonds during the atom generation procedure. After decoding, we propose to use a space matching method to limit the output search space, and also validate the output sequence using the RDKIT library.

### A. Representation Learning on Molecular Graphs

In this section, we introduce three methods for graph representation learning. The comparison of three methods will show that EAGCN is a suitable way to learn atom feature vectors and the edge feature vectors.

We denote a graph as $G = (V, E)$, where $V$ is a set of nodes with $|V| = N$, and $E \subseteq V \times V$ is a set of edges with $|E| = M$. An adjacency matrix $A$ is a square binary matrix. $X$ is a feature matrix, where the $i$-th row represents the feature vector of node $i$ and the $j$-th column is the vector of feature $j$ for all the nodes. In addition, the edges in the graph have $K$ number of possible edge attributes. For the layer $l$, 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 $i$. Here $F$ is the number of features in each node. When $l$ is equal to 1, the input feature matrix $H^1$ is $X$. The linear transformation from the input of the layer $l$ to its output is parameterized by matrix coefficients $\{W_k^l \in \mathbb{R}^F \times \mathbb{R}^{F'_k} | 1 \leq k \leq K\}$.

*1) Traditional Graph Convolutional Networks (GCN).:* Graph convolution networks are proposed [21] to learn representations of the nodes in a graph. Each hidden layer in GCN is formulated as:

$$f(H^l) = \sigma(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}H^l W^l), \tag{1}$$

where D is the diagonal node degree matrix. The neighboring information is aggregated for each layer and more global information is learned with deeper layers. However, only the node representation is learned in GCN and the important edge information is missing.

*2) Message Passing Neural Networks (MPNN).:* Message Passing Neural Networks is a variation of GCN in that it leverages the assumption of the continuous edge features and introduce a two-phase passing scheme. The learning phase is defined as:

$$\begin{aligned} m_n^{l+1} &= \sum_{u \in ne(n)} M^l(h_n^l, h_u^l, e_{(n,u)}), \\ h_n^{l+1} &= U^l(h_n^l, m_n^{l+1}) \\ e_{(n,u)}^l &= E_l(h_n^l, h_u^l, e_{(n,u)}^l). \end{aligned} \tag{2}$$

The third equation employed edge update [8] and can learn the edge representation of molecule graph. The continuous edge property is suitable to encode distance information in a molecule graph, but it cannot provide explainable information for multiple edge features and it cannot handle relations between multiple graphs.

Since MPNN with edge embedding also contains two views (atoms and edges), we also apply our multi-view attention model on MPNN. However, the atom embedding is vector for single atom, thus only intra atom/edge attention is applied.

*3) Edge Attention based GCN (EAGCN).:* The edge attention graph convolution layer (EAGCN) [9] provides a promising way to learn multiple relational strengths (views) of node interactions with neighbors using the edge attributes. Sharing the attention weights across different molecular graphs helps to learn the inherently invariant properties in multiple graphs. Meanwhile, with the edge attention, more reasonable node representations are generated which aggregates neighboring information based on node-to-node interactions. Each EAGCN layer generates both node and edge representations, as is illustrated in Fig 1.

All the datasets used in the paper have $K = 6$ edge attributes. Each attribute has several discrete values, which means each attribute has different edge types. For edge attribute $i$, all the learnable attention weights assigned for edge types are grouped as an edge attention weighted adjacency dictionary $\mathbf{D}_k^l$ as shown in Figure III-A3. We have $K$ different edge attributes for the dataset, as shown in Table I. If the edge feature contains $d_k$ discrete values for the edge attribute $k \in K$, EAGCN creates a dictionary $\mathbf{D}_k^l \in \mathbb{R}^{d_k}$ for modeling the strengths of interaction for edge attribute k in layer $l$. The weights $\{\alpha_{k,j}, 1 \leq j \leq d_k\}$ in dictionary $\mathbf{D}_k^l$ will be learned by our model. For all the edge attributes, a set of dictionaries will be created as $\{\mathbf{D}_1^l, ..., \mathbf{D}_K^l\}$, which is not only shared for one graph but also used for all the graphs in the dataset.

Using the dictionary set, EAGCN obtains a set of weighted adjacency matrices $\{A_{att,k}^l, 1 \leq k \leq K\}$ corresponding to multiple edge attributes. The weight $\alpha_k^l$ for edge $e$ in edge attention weighted matrix $A_{att,k}^l$ is obtained by lookup table operations illustrated in [9]. By combining all the edge attention matrices $\{A_{att,1}^l, ..., A_{att,K}^l\}$, we get the edge representations tensor $\mathbb{A} \in \mathbb{R}^{n \times n \times K}$. Hence each **edge representation** is a $K$ dimensional vector in $\mathbb{A}$, which will be learned during the backpropagation. Then **node representations** will be updated and generated using the edge attention weighted matrices based multi-view graph convolutional layer as below.

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

$$R_k^{l+1} = \sigma(A_{att,k}^l H^l W_k^l), \tag{3}$$

for $1 \leq k \leq K$, where $\sigma$ is an activation function. After computing, we get a set $\{R_k^{l+1} \in \mathbb{R}^N \times \mathbb{R}^{F'_k} | 1 \leq k \leq K\}$. Then the node feature matrix $R^{l+1}$ is the concatenation of all

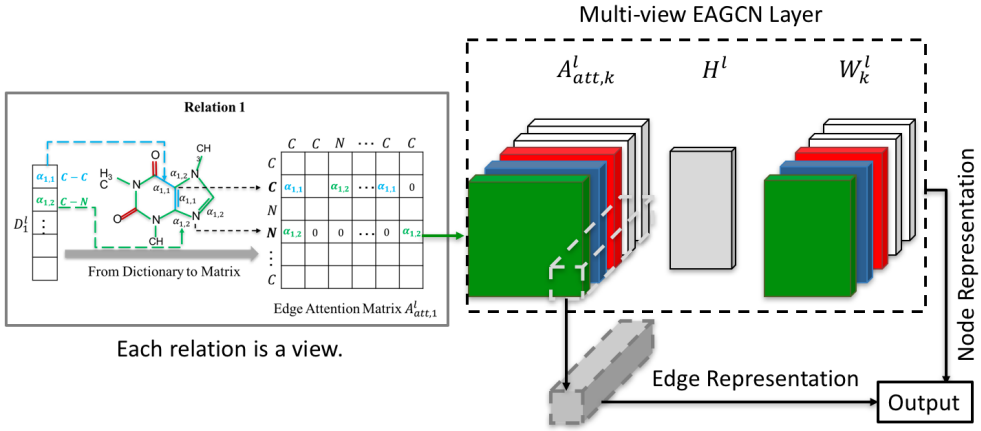


Fig. 1. The edge attention based multi-view graph convolutional layer for node and edge representation learning. Attention dictionary for relation or view $k$ is defined based on the $k^{\text{th}}$ edge attribute. Attention matrix is then formed from the values in the dictionary. The $k$ generated matrices $A^l_{att,k}$ are fed to EAGCN layer. The output of the model is a set $H^l_k$ of node representations and a set of multi-view edge representations. For the bond between node $p$ and node $q$, the edge representation is described as a vector $[A^l_{att,1}[p][q], A^l_{att,2}[p][q], ..., A^l_{att,k}[p][q]]$, where $k$ is the number of views.

the items in this set:

$$R^{l+1} = [R_1^{l+1}, R_2^{l+1}, ..., R_K^{l+1}]. \tag{4}$$

### B. Nested Seq2Seq Model

In order to translate from the embedded atoms to the sequence of the products, we propose a nested attention based Seq2Seq for atom representation and edge representation. The Seq2Seq model consists of two distinct recurrent neural networks (RNN): (1) an encoder that processes the input vector and outputs its representation, and (2) a decoder that uses this representation to output a probability over a prediction. For these two RNNs, we apply the long short-term memory (LSTM) [22] considering the potential length of the molecule and the ability to handle long-range relations in sequences. An LSTM consists of units that process the input data sequentially. Each unit at each time step $t$ processes an element of the input $x_t$ and the network's previous hidden state $h_{t-1}$. In our model, since both the atom and edge representations are fed to an RNN, we use two separate LSTM units to learn and update atom representation and edge representation simultaneously. As a consequence, there will be two outputs for each recurrent state and both of them will be fed to the attention network. When predicting the product, the decoder will pay attention to not only the atom information but also on edge information. The structure of the network can be viewed in Figure 2.

Nodes and edges use a different set of parameters but in a nested way. The output and the hidden state transition of the representation are defined by:

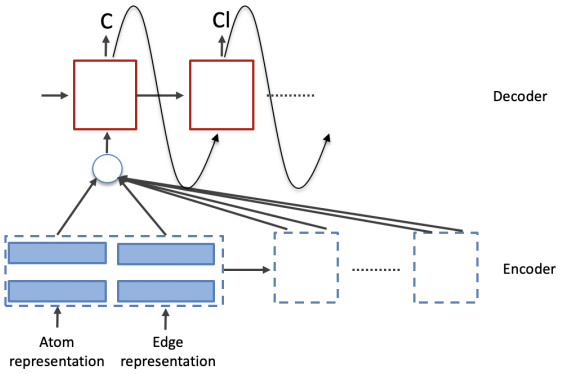


Fig. 2. The Nested Seq2Seq model. Atom and edge representations provide two views to the Seq2Seq model. The decoder will focus on both information and decides the reaction sites.

$$\begin{aligned}
i_t^g &= \sigma(W_i^g \cdot x_t^g + U_i^g \cdot h_{t-1}^{\bar{g}} + b_i^g),\\
f_t^g &= \sigma(W_f^g \cdot x_t^g + U_f^g \cdot h_{t-1}^{\bar{g}} + b_f^g),\\
o_t^g &= \sigma(W_o^g \cdot x_t^g + U_o^g \cdot h_{t-1}^{\bar{g}} + b_o^g),\\
c_t^g &= f_t^g \times c_{t-1}^{\bar{g}} + i_t^g \times tanh(W_c^g \cdot x_t^g + U_c^g \cdot h_{t-1}^{\bar{g}} + b_c^g),\\
h_t^g &= o_t^g \times tanh(c_{t-1}^{\bar{g}}),
\end{aligned} \tag{5}$$

where $g \in \{a, e\}$. Note that $x_t^a$ denotes the $t^{\text{th}}$ element encoded by the output of the final output of EAGCN. $i_t^g$, $f_t^g$ and $o_t^g$ are the input gates, forget gates, and output gates, respectively; $c_t^g$ is the cell state vector; $W^g$, $U^g$ and $b^g$ are model parameters learnt during training; $\sigma$ is the sigmoid function. In order to capture both the forward and backward correlations of a SMILES string, we used a bidirectional LSTM (BLSTM). A BLSTM processes the input sequences in both directions, so they have context not only from the past but also from the future. $\overrightarrow{h_t^g}$ and $\overleftarrow{h_t^g}$ represent the forward

and backward processes, respectively. The hidden states of a BLSTM are defined as: $h_t^g = \{\overleftarrow{h_t^g}, \overrightarrow{h_t^g}\}$.

The encoder learns a representation of both the atoms and edges, and can be formalized as $Encode^g = f(W_e \cdot x_t^g, h_{t-1}^g)$.

Note that the input atom representation has multiple view. In the Seq2Seq model, we use separate sets of weights for different views. And for edge representation, the each edge is represented by a singe vector, so only one set of weights are adopted in the Seq2Seq model. The decoder predicts the probability of observing an outcome $\hat{y} = \{\hat{y}_1, ..., \hat{y}_M\}$:

$$P(\hat{y}) = \Pi_{i=0}^{M} p(\hat{y}_i | \{\hat{y}_1, ..., \hat{y}_{i-1}\}). \tag{6}$$

### C. Luong's Attention Mechanism

Observe that the edge attention and node attention are calculated separately, and both the results are concatenated and fed to the decoder. Similar to Seq2Seq model, we also adopt Luong's Attention Mechanism where the context vector is computed by the attention factor:

$$\begin{aligned} \alpha_{it}^g &= \frac{\exp(s_i^g \cdot W_\alpha^g \cdot H_t^g)}{\sum_{t=0}^{T} \exp(s_i^g \cdot W_\alpha^g \cdot H_t^g)} \\ c_i^g &= \sum_{t=0}^{T} \alpha_{it}^g \cdot H_t^g, \ and \ c_i = [c_i^a | c_i^e]. \end{aligned} \tag{7}$$

The attention factor is defined as the concatenation of the two views and is used to generate the attention vector through a single layer neural network:

$$a_i = tanh(W_b \cdot [c_i^a | c_i^e]). \tag{8}$$

This layer learns the function choosing from the important attention of atoms or bonds. Note that both $W_\alpha$ and $W_b$ are learned weights. Then the attention factor can be used to compute the probability for a particular output distribution:

$$p(y_i | \{y_1, ..., y_{i-1}\}, a_i) = softmax(W_p \cdot [a_i]). \tag{9}$$

### D. Multi-view Attention Mechanism

The attention factor in equation 7 can only capture the single view properties. However, EAGCN provides much more meaningful multi-view information, which should be utilized carefully. In this subsection, we introduce self attention to model correlation between different views inside each atom or edge and mutual attention to combine information between atoms and edges.

*1) Self Attention Inside each Atom or Edge:* Self attention in each atom/edge is important in chemical reaction process since the combination of different view may result in different chemical properties. For example, a carbon atom in a ring will perform very differently with a carbon atom in a triple bond. So the correlation between views will have significant impact on the attention score.

To aggregate multiple views for a single atom or single edge, we use trainable correlation matrix to act as correlation coefficiencies, which will be further utilized to calculate the final attention score in Seq2Seq model. We adopt two way attention [23] to model the such correlation. The inter Atom attention is defined as:

$$\sigma_e^{inter} = tanh((H^e)^T W^e H^e), \tag{10}$$

and

$$\Sigma_a = tanh((H^a)^T W^a H^a), \tag{11}$$

Where $W_e$ and $W_a$ are weight correlation matrices to be learned. Note that $W^a$ and $W_e$ have the same dimension $\mathbb{R}^K \times \mathbb{R}^K$ but the output will be different because of the difference of atom representation and edge representation. $\sigma_e^{inter}$ is a scalar while $\sigma_a \in \mathbb{R}^{F_k'} \times \mathbb{R}^{F_k'}$.

To reduce the dimension of $\Sigma_a$, instead of using computationally intensive method as in [24], we use light weight aggregation method similar to [25] to squeeze the scoring matrix to a scalar. Here, we pass the atom attentions to a single layer neural network. Different from [25], we take the operations of both max pooling and average pooling to generate the self attention scores. The max pooling is to find the view that have the maximum impact while average pooling is to average the impact from all the views.

$$\begin{aligned} &\sigma_a^0 = \sigma(W_a \Sigma_a), \ \sigma_a^1 = softmax(max(\Sigma_a)) \\ &\sigma_a^2 = softmax(max(\Sigma_a^T)), \ \sigma_a^3 = softmax(ave(\Sigma_a)) \\ &\sigma_a^4 = softmax(ave(\Sigma_a^T)), \end{aligned} \tag{12}$$

where $g \in a, e$, $max$ and $ave$ are column-wise operations. The final score is calculated as:

$$\sigma^{final} = softmax(\sum_{0}^{4} \lambda_i \sigma^i), \tag{13}$$

where $\lambda_i$ are hyperparameters of those five kind of attentions.

*2) Mutual Atom-Edge Attention:* Mutual Atom-Edge attention is to incorporate correlation between atom views and edge views. Such correlation is critical in chemical reaction since the strength of a edge can be significantly impacted by another atom. So the atom-edge interaction must be carefully considered. Due to the dimension mismatch between atom and edge representation, direct combination is not achievable. The attention score is defined as:

$$\sigma^{cross} = tanh((H^e)^T W_{ea} f(H^a)), \tag{14}$$

where $W_{ea}$ is a $\mathbb{R}^K \times \mathbb{R}^K$ correlation matrix and $f(H^a)$ is a squeezing function that reduce the dimension of $H_a$ to $\mathbb{R}^{F_k'} \times 1$. Similar to equation 12 typical choices of $f$ can be max-pooling, average-pooling, or a simple matrix multiplication. The calculation is the same as is described in previous section.

### E. Output Validation

In sequence generation RNN, the output length and correctness are out of control since the search space of the sampling process of the decoder is unbounded. We use several validation techniques to bound the output search space. The space of a molecule is defined as $M$ with size of $N$. The input space is a subspace of a molecule $I^K \in M$, where $K$ is the length

of an input atom. The output space is $O^L$, which is different from the input since the length of the output is different from that of the reactants. Given two input molecules in $I^{K_1 \times K_2}$, we can further limit the output space to $O_L \in I^{K_1 \times K_2}$, and $L \leq K_1 + K_2$. During the decoder sampling process, we can reduce the search space for $\hat{y}_t$ at step $t$ as:

$$O^t \in I^{K_1 \times K_2} / O^{t-1}. \tag{15}$$

To further validate the output, we use RDKit to check the correctness during the beam search process. The output distribution can be further formalized as:

$$p(y_i | \{y_1, ..., y_{i-1}\}, c_i) = v_{rd} \cdot softmax(W_p \cdot a_i) \tag{16}$$

where $v_{rd}$ is the output of the RDkit validation check. Note that $v_{rd}$ is a probability indicating if the molecule is valid. The lower this probability, the less likely that an atom will be chosen in the beam search ranking process. However, there is still a chance that a new molecule that has never been seen before will be selected as the output.

## IV. Empirical Evaluation

In this section, we evaluate our pipeline on two commonly evaluated datasets, Lowe's grants dataset and Jin's USPTO dataset. We compare our pipeline with WLDN [12] and pure Seq2Seq [10] models. To prove the efficiency, we also compare it with different preprocessing steps: (1) One Hot Embedding, (2) GCN with atom embedding, (3) MPNN with both atom and edge embedding, and (4) EAGCN with both atom and edge embeddings. To verify the effectiveness of multi-view learning, we evaluate different combinations of views in EAGCN and discuss the insights gained from experiments.

### A. Experimental Setup

The node features and edge attributes are extracted using the RDKit. We converted SMILES into ".mol" format, which contains the molecular structure information used to build the molecular graph. The input for EAGCN is the molecular graph and then mapped back to SMILES strings. In order to apply a fixed-size representation, for the atom pair types whose frequencies are lower than the threshold, we will set the same attention weight for them in the dictionary. Ten independent runs with different random seeds are performed and the averages are reported. We use the adaptive moment (ADAM) optimization algorithm for training the model.

The training and evaluation processes are as follows: 1) Pre-train EAGCN, GCN, and MPNN using QM-9 dataset. The dataset contains 134K molecules made up of CHONF. Note that the number of heavy atoms in the chemical reaction is larger than that in QM9. After pretraining, 2) Connect the output layer of representation learning network to decocer model. The RNN scan the output of the representation learning model based on the order of SMILES in both directions and learn the hidden state. During this procedure, Some edge representations are ignored since SMILES cannot describe all the graph geometric information. However, by employing two EAGCN layers, the information of the one-hop neighbors is included in each edge, so that it covers all the uncovered edges when traversing SMILES. 3) All the networks are then connected as proposed and jointly trained. 4) The final model is tested 10 times. We pick the network with the highest accuracy to be the final candidate. We take the output of the candidate representation learning model and feed it to the Bi-modal decoder network, train the decoder network 10 times and output the average accuracy. 5) The beam search hyperparameter can be limited to 6 thanks to the output validation process. 6) We adopt the full-sequence accuracy, where a test prediction is considered correct only if all the tokens are identical to the ground truth. The models have been implemented using PyTorch and run on Ubuntu Linux 16.04 with NVIDIA Titan RTX Graphics Processing Units.

TABLE I
EDGE ATTRIBUTES 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). |
| `Atom Pair Distance` | Discrete distances through Gaussian basis function. |
| `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. |

### B. Edge Attributes

To employ full information of the chemical bond, we include six different attributes, as shown in Table I. Atom pair type reflects the basic bond energy defined by the atoms. Atom pair distance, combined with atom pairs, describes the atom-wise force. Bond order reflects the general bond strength. Aromaticity, Conjugation and Ring Status reflect the special structure which may not be revealed by the SMILES string.

### C. Datasets

As mentioned in [10], all the openly available chemical reaction datasets were derived from the patented text-mining work of Daniel M. Lowe. What makes the dataset particularly interesting is that the quality and noise correspond well to the data a chemical company might own. The granted patent is made of 1,808,938 reactions, which are described using SMILES. The dataset is incomplete and contains noise and errors. It is not suitable for direct training. To evaluate our model and compare it with the existing models, we use two reduced pre-processed datasets as reported in [10] and [12].

### D. Attention Factor

To show the effectiveness of the graph model, we compare the attention factor between the final model and all the candidates. Note that the correctness of the attention factor in the decoder model is important and to some extent dominates the correctness of the final output. In this section, we show that the attention factor is predicted correctly when with edge presentation input and with the correct attention on edges, the

TABLE II
ACCURACY RESULTS ON TWO COMMONLY USED DATASETS

| | Jin's USPTO | | | | Lowe's | | |
|---|---|---|---|---|---|---|---|
| Model | top-1 | top-2 | top-3 | top-5 | top-1 | top-2 | top-3 |
| WLDN | 74.0 | N/A | 86.7 | 89.5 | N/A | N/A | N/A |
| Seq2Seq | 80.3 | 84.7 | 86.2 | 87.5 | 65.4 | 71.8 | 74.1 |
| GCN+Seq2Seq | 80.8 | 85.6 | 86.4 | 87.9 | 65.9 | 72.3 | 75.8 |
| MPNN /o edge+Seq2Seq | 80.7 | 85.9 | 86.5 | 87.7 | 66.3 | 72.1 | 76.1 |
| MPNN /w edge+Seq2Seq | 86.8 | 90.0 | 92.2 | 93.3 | 74.3 | 80.1 | 83.4 |
| MPNN /w edge+Seq2Seq+intra attention | 87.5 | 90.9 | 93.1 | 94.5 | 75.2 | 80.8 | 84.3 |
| EAGCN+Seq2Seq+single att | **88.1** | **92.3** | **94.4** | **95.6** | **76.2** | **81.2** | **84.8** |
| EAGCN+Seq2Seq+multiview att | **89.7** | **93.8** | **96.0** | **96.7** | **78.8** | **83.1** | **87.4** |

product of the chemical reaction is then correctly induced. The auxiliary notations or the connectivity notations(such as "(" or ")") are ignored in the illustration but is actually included in the original model. From figure IV-D, we see that without edge embedding (GCN and MPNN), the reaction center is not predicted with high confidence since they focus on the irrelevant parts of the input. Also observe that for EAGCN without edge embedding, even though the node attention provides some information of the edges to the nodes, the attention of the decoder is not correctly focused. With edge information provided, the model pay some attention to the reaction center and are more focused compared to models without edge attention. Here more focus means less distraction by other atoms when building the new connection of the true reaction center. Moreover, multi-view achieve the most focus and accurate attention then all the others.

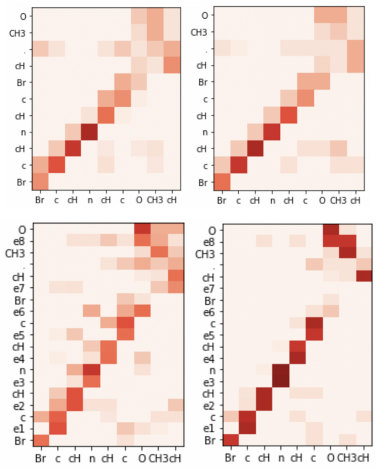

Fig. 3. Visualization for attention factor. Top left: GCN+Seq2Seq, Top right: MPNN+Seq2Seq, Bottom left: EAGCN+Seq2Seq, Bottom right: EAGCN+Seq2Seq+Multiview attention

Another phenomenon is that when selecting the output atom, the attention is paid on not only the atom from the input but also the edge representation after the previous atom and before the next candidate atom, indicating that the decoder understands the mechanism on locating the reaction center.

### *E. Comparison Results*

We first show the results for Jin's pre-processed dataset in Table II in columns 2 to 5. The results are categorized into two groups. Group 1 includes the preprocessing without edge embedding, i.e., GCN and MPNN. Group 2 includes MPNN and EAGCN with edge embedding. Note that since Jin's dataset is well cleaned with less noise, all the models can achieve a high accuracy. Models with edge embedding achieve state-of-the-art prediction accuracy. In this Table, we also show the results on Lowe's dataset in columns 6 to 8. As we see, the average accuracy of with edge embedding is significantly better than only atom embedding. This indicates that atom representation alone is not enough to describe the property of the bond and thus not helpful to find the reaction center. With edge embedding, both atom and edge representations are output and being considered in the inference attention step, thus the overall prediction accuracy is improved.

Specifically, EAGCN with edge embedding slightly outperforms MPNN in that the multi-view model learns more information from the graph than directly from message passing. As is shown in table II, edge embedding models outperform the state-of-the-art on both datasets, and the improvement margin of Lowe's dataset is higher than Jin's dataset. This shows that the model can handle complex and noisy datasets.

Moreover, we compare the results with single attention and multi-view attention on MPNN and EAGCN. In MPNN, we use intra atom and edge attention only. Both model with multi-view attention perform a clear improvement in all cases. And the improvement in EAGCN is higher than the improvement in MPNN, indicating the effectiveness of considering both inter and intra correlations.

TABLE III
RESULTS OF DIFFERENT VIEWS OF EAGCN.

| | Jin's USPTO | | | Lowe's | | |
|---|---|---|---|---|---|---|
| Model | top-1 | top-3 | top-5 | top-1 | top-2 | top-3 |
| View1 | 82.5 | 87.6 | 89.7 | 71.2 | 78.3 | 81.3 |
| View2 | 80.1 | 89.8 | 91.3 | 70.2 | 77.7 | 80.9 |
| View3 | 78.2 | 82.5 | 83.3 | 62.8 | 69.6.1 | 72.6 |
| View4 | 86.7 | 93.7 | 94.8 | 75.1 | 80.9 | 84.1 |
| All views | **88.1** | **94.4** | **95.6** | **76.2** | **81.2** | **84.8** |

### F. Multi-view Analysis

We further run experiments showing the effectiveness of multi-view learning. We modify the EAGCN model with view of 1) atom pair only, 2) atom distance only, 3) bond type only, 4) atom pair, distance, and bond order, and 5) all views. We show the results in Table III. All the single view representation learning methods failed to achieve a high accuracy. Moreover, the view with only bond type performs significantly worse than the other views. On the other hand, the single view with only atom pair outperforms other single view models. When comparing view4 (atom pair, distance, and bond order) with full views model, the accuracy drop is moderate. Views of aromaticity, conjugation, and ring status contain partial structural information with a moderately significant impact on the outcome prediction.

## V. Conclusions: Significance and Impact

In this paper we address an important problem in Materials Genomics, i.e., that of predicting the outcomes of chemical reactions. To encode molecules with full information, we employ different graph embedding methods: graph convolution, message passing network(MPNN) and multi-view edge attention graph convolution(EAGCN) model to learn both node and edge representations. To further employ the multi-view embedding information, we propose self and mutual attention method working with MPNN and EAGCN. Compared with the pure Seq2Seq model, our model includes the edge information which is important in predicting the reaction centers. Compared with single attention models, the attention factor of our method incorporate correlations among different aspects, providing higher confidence attention scores. Empirical results reveal that the end to end pipeline achieves a superior accuracy compared to all the algorithms that have been published in the literature for the same problem. We also believe that the paradigm we have introduced in this paper is of independent interest in the machine learning domain in general.

## References


[1] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[2] Mikhail Drobyshevskiy, Anton Korshunov, and Denis Turdakov. Learning and scaling directed networks via graph embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 634–650. Springer, 2017.

[3] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *stat*, 1050:7, 2017.

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

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

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

[7] 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. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2016.

[8] Peter Bjørn Jørgensen, Karsten Wedel Jacobsen, and Mikkel Nørgaard Schmidt. Neural message passing with edge updates for predicting properties of molecules and materials. In *32nd Conference on Neural Information Processing Systems*, 2018.

[9] Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. Edge attention-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1802.04944*, 2018.

[10] Philippe Schwaller, Theophile Gaudin, David Lanyi, Costas Bekas, and Teodoro Laino. " found in translation": Predicting outcome of complex organic chemistry reactions using neural sequence-to-sequence models. *arXiv preprint arXiv:1711.04810*, 2017.

[11] Andrea Cadeddu, Elizabeth K Wylie, Janusz Jurczak, Matthew Wampler-Doty, and Bartosz A Grzybowski. Organic chemistry as a language and the implications of chemical linguistics for structural and retrosynthetic analyses. *Angewandte Chemie*, 126(31):8246–8250, 2014.

[12] Wengong Jin, Connor Coley, Regina Barzilay, and Tommi Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. In *Advances in Neural Information Processing Systems*, pages 2604–2613, 2017.

[13] Juno Nam and Jurae Kim. Linking the neural machine translation and the prediction of organic chemistry reactions. *arXiv preprint arXiv:1612.09529*, 2016.

[14] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, pages 6533–6542, 2017.

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

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

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

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

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

[20] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[21] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[23] Cicero dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*, 2016.

[24] Ye Yuan, Guangxu Xun, Fenglong Ma, Yaqing Wang, Nan Du, Kebin Jia, Lu Su, and Aidong Zhang. Muvan: A multi-view attention network for multivariate temporal data. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 717–726. IEEE, 2018.

[25] Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Nan Du, Wei Fan, Kai Lei, and Ying Shen. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6318–6325, 2019.