PAPER

Chemical Reaction Enhanced Graph Learning for Molecule Representation

Anchen Li,[1] Elena Casiraghi[1,2,3,4] and Juho Rousu[1]

[1]Department of Computer Science, Aalto University, Finland, [2]AnacletoLab, Dipartimento di Informatica, University of Milan, Italy, [3]Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA and [4]ELLIS, European Laboratory for Learning and Intelligent Systems, Milan Unit, Italy

Abstract

Motivation: Molecular representation learning (MRL) models molecules with low-dimensional vectors to support biological and chemical applications. Current methods primarily rely on intrinsic molecular information to learn molecular representations, but they often overlook effectively integrating domain knowledge into MRL.
Results: In this paper, we develop a reaction-enhanced graph learning (RXGL) framework for MRL, utilizing chemical reactions as domain knowledge. RXGL introduces dual graph learning modules to model molecule representation. One module employs graph convolutions on molecular graphs to capture molecule structures. The other module constructs a reaction-aware graph from chemical reactions and designs a novel graph attention network on this graph to integrate reaction-level relations into molecular modeling. To refine molecule representations, we design a reaction-based relation learning task, which considers the relations between the reactant and product sides in reactions. In addition, we introduce a cross-view contrastive task to strengthen the cooperative associations between molecular and reaction-aware graph learning. Experiment results show that our RXGL achieves strong performance in various downstream tasks, including product prediction, reaction classification, and molecular property prediction.
Availability and implementation: The code is publicly available at https://github.com/coder-ACAC/RLM
Contact: anchen.li@aalto.fi. **Supplementary Information:** Supplementary data is available at Bioinformatics online.

Introduction

Molecule representation learning (MRL) techniques are crucial for combining machine learning with biological and chemical sciences (Yi *et al.*, 2022). MRL encodes molecules as low-dimensional vectors. These vectors retain molecule information, facilitating their use as features in downstream applications (e.g., product prediction, reaction classification, and molecular property prediction). A variety of MRL methods have been proposed, which roughly fall into the following two categories.

One school is SMILES-based methods (Fabian *et al.*, 2020), which utilize SMILES strings as input and employ natural language models as their base architectures. However, they struggle with capturing molecule structures. The other school treats molecule topology as a graph, and models molecules with graph neural networks (GNNs) (Xu *et al.*, 2021). Although GNN-based methods generally outperform SMILES-based ones, they typically focus on designing GNN architectures, neglecting the efficient integration of domain knowledge.

Recent studies (Wang *et al.*, 2022a) use chemical reactions as domain knowledge for MRL. Typically, reactions are represented by equations, with reactants on the left side and products on the right (cf. Definition 2 in Section Preliminaries). These methods first learn molecule embeddings from molecular graphs and then optimize embeddings by equating the sum of reactant embeddings with the sum of product embeddings for each reaction. Despite effectiveness, we argue that they face at least one of the following issues.

Firstly, these reaction-based methods treat molecules as isolated data instances and rely solely on molecule structures for representation, which ignores the insights from molecule relations inherent in chemical reactions. For example, molecules involved in the same reaction (as reactants/products) may exhibit greater similarities and correlations with each other than with molecules from different reactions. To illustrate this reaction-related relation, we construct a reaction-aware graph (cf. Definition 4 in Section Preliminaries) based on a reaction set, as shown in Fig.1(a). In this graph, nodes are molecules and edges denote molecule relations driven by reactions. For molecule A, its first-order neighbors (molecules F, G, and H) represent products that can be derived from A through reactions. Molecule A's second-order neighbors (molecules B, C, and D) suggest a property/structure similarity with A, inferred from shared reaction products. Moreover, molecule B is likely more similar to A than C or D, as evidenced by a greater overlap in the reaction products. These analyses inspire us to consider the potential benefits of incorporating molecule relations from the reaction-aware graph into MRL.

Secondly, these methods ignore the transformation relation learning between reactants and products. Their assumption that the summed embeddings of reactants and products

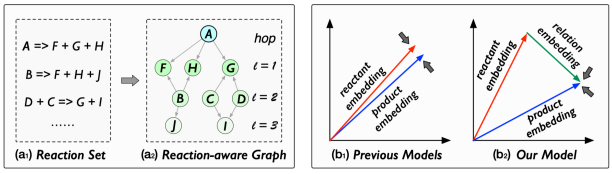


Fig. 1. (a_1) and (a_2): An illustration of the reaction-aware graph. (b_1) and (b_2): Comparisons of previous models and our model in considering reactant-product relations of the chemical reaction.

should be equal (as shown in Fig.1(b_1)) essentially reduces all reactions to an identity transformation, which oversimplifies the complexity of chemical processes. In reality, reactions involve various changes, such as the number of bonds (e.g., breaking old bonds and forming new ones) and energy variations (e.g., endotherms and exotherms) before and after the reaction. The assumption in current studies fails to model these changes.

Motivated by these gaps, we introduce a reaction-enhanced graph learning framework (RXGL) for MRL. In the molecule modeling stage, we design dual graph learning modules. The first module utilizes graph convolutions on molecular graphs to capture the structural information of molecules. The second module first involves a reaction-aware graph and then creates a GNN to extract reaction-level molecular relations for molecule feature learning. In the optimization stage, we introduce a reaction-based relation learning method that considers the relation between reactants and products in chemical reactions. Specifically, we employ a memory network (Miller *et al.*, 2016) to learn a latent relation vector that connects reactant and product embeddings (as shown in Fig.1(b_2)). Through the delicate key and memory components in this network, the learned relation vectors could capture the hidden semantic correlations between the reactant and product in each reaction. Furthermore, we incorporate a cross-view contrastive learning task to enhance molecule representations in dual graph modeling. This task treats the molecular and reaction-aware graphs as distinct yet correlated views. By employing contrastive learning, we capture cooperative associations between views, thereby integrating their agreements into molecule representations.

Our contributions are summarized as follows:

- We propose a GNN-based MRL framework RXGL, which introduces the chemical reaction-aware graph to assist in learning molecule representations.
- We devise a reaction-based relation modeling approach, which learns the relations between reactant and product sides in chemical reactions to guide MRL.
- We design a cross-view contrastive learning task to enhance the cooperative association between the molecular graph and reaction-aware graph modeling views.
- Experiment results show that molecule embeddings learned by our RXGL benefit various tasks, i.e., product prediction, reaction classification, and molecular property prediction.

Preliminaries

This section introduces notations and formulates the problem.

Notations

Our method RXGL uses the molecular graph and reaction-aware graph for molecule representation learning. We first introduce the definition of the molecular graph:

Definition 1. Molecular Graph. Given a molecule set $\mathcal{M}$, each molecule $m \in \mathcal{M}$ has a graph structure $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$, which includes an atom node set $\mathcal{V}_m$ and a bond edge set $\mathcal{E}_m$. Each atom $a_i \in \mathcal{V}_m$ is represented by a vector encoding its features, and bond $b_i \in \mathcal{E}_m$ is represented by its bond type.

The reaction-aware graph is constructed based on chemical reactions. The chemical reaction is defined as follows:

Definition 2. Chemical Reaction. Consider a molecule set $\mathcal{M}$ and a reaction set $\mathcal{X}$, a reaction $x_i \in \mathcal{X}$ defines a transformation from a reactant set $R_i \subset \mathcal{M}$ to a product set $P_i \subset \mathcal{M}$, denoted as $x_i : R_i \rightarrow P_i$, where sets $R_i = \{r_{i_1}, r_{i_2}, \ldots\}$ and $P_i = \{p_{i_1}, p_{i_2}, \ldots\}$ represent the reactants and products involved in reaction x_i, respectively.

According to the definition of the chemical reaction, we define the reaction-level relation between molecules:

Definition 3. Reaction-level Molecule Relation. For a given reaction $x_i : R_i \rightarrow P_i$, a reaction-level molecule relation is defined between any reactant $r_{i_a} \in R_i$ and product $p_{i_b} \in P_i$.

Based on the reaction and reaction-level molecule relation, the reaction-aware graph is defined as follows:

Definition 4. Reaction-aware Graph. A reaction-aware graph is defined as $\mathcal{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$, where $\mathcal{V}_x = R \cup P$ is the node set, and $\mathcal{E}_x \subset R \times P$ is the edge set including reaction-level molecule relations between reactants R and products P.

Fig.1(a) shows an example of the reaction-aware graph. It is worth noticing that the construction of the reaction-aware graph draws inspiration from the metabolic networks (Wagner and Fell, 2001) and chemical reaction networks (Wen *et al.*, 2023) in synthetic biology. Its structural design facilitates the modeling of reaction-level molecule relations.

Problem Formulation

Given the molecule m_i's graph structure $\mathcal{G}_{m_i}$, reaction-aware graph $\mathcal{G}_x$ and a chemical reaction set $\mathcal{X}$, we aim to learn m_i's representation $\mathbf{h}_i$. Then, learned molecule representation $\mathbf{h}_i$ can be applied to various downstream tasks.

Reaction-enhanced Graph Learning (RXGL)

This section presents our RXGL method, as shown in Fig.2. We first introduce dual graph learning modules to model molecule representations. Then, we propose a reaction-based relation learning task and a cross-view contrastive learning task to optimize molecule representations.

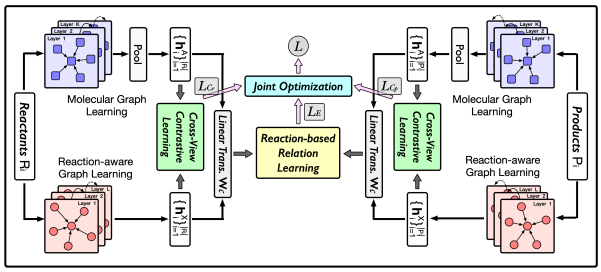


Fig. 2. The framework of our RXGL method.

Molecule Representation Modeling

We introduce dual modules: a molecular graph learning module and a reaction-aware graph learning module. These modules are designed to integrate atom-level and reaction-level features into molecule representations, respectively.

Molecular graph learning module

This module first uses GNNs to model atoms and then utilizes pooling operations on the molecular graph to inject atom-level

features into molecule representations, as shown in Fig.3(a). For molecule m's graph structure $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$, we model representation $\mathbf{a}_i^k$ of atom $a_i \in \mathcal{V}_m$ in the k-th GNN layer:

$$\mathbf{a}_i^k = \text{Aggregate}(\mathbf{a}_j^{k-1} | a_j \in \mathcal{N}_i \cup a_i), \tag{1}$$

where $\mathcal{N}_i$ is the atoms connected to atom a_i in graph $\mathcal{G}_m$. The choice of Aggregate function is the key to designing GNN, leading to the proposal of various structures (Li *et al.*, 2022, 2024). We utilize four common GNNs (i.e., GCN (Kipf and Welling, 2017), GAT (Veličković *et al.*, 2018), SAGE (Hamilton *et al.*, 2017), and TAG (Du *et al.*, 2017)) for this module. We introduce these GNNs in the supplementary materials.

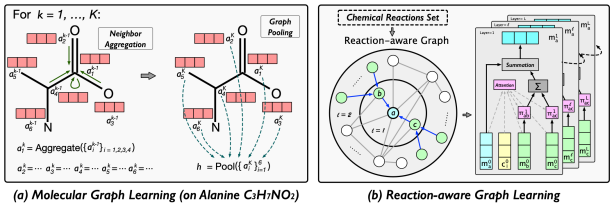


Fig. 3. Molecular and reaction-aware graph learning.

To describe initial feature $\mathbf{a}_i^0$ of atom a_i in $\mathcal{G}_m$, we use four types of atom properties: element type, charge, presence in an aromatic ring, and the number of attached hydrogen atoms. Each type of property is represented as a one-hot embedding, and four embeddings are concatenated as the initial atom feature. Note that we do not represent bond types with explicit feature vectors because the bond type could be inferred by the connected atom features (Wang *et al.*, 2022a).

By stacking K GNN layers, a pooling function is used to generate atom-level molecule representation $\mathbf{h}^A$, as follows:

$$\mathbf{h}^A = \text{Pool}\left(\mathbf{a}_i^K | a_i \in \mathcal{V}_m\right) = \sum\nolimits_{a_i \in \mathcal{V}_m} \mathbf{a}_i^K. \tag{2}$$

Reaction-aware graph learning module

This module distills reaction-level relations from the reaction-aware graph to model molecule representations, as shown in Fig.3(b). We first construct a reaction-aware graph $\mathcal{G}_x = (\mathcal{V}_x, \mathcal{E}_x)$. To describe initial feature $\mathbf{m}_i$ of molecule $m_i \in \mathcal{V}_x$ in $\mathcal{G}_x$, we use the molecule functional group information. We first use a one-hot embedding to denote each functional group, and then the molecule is represented by the summation of its functional group embeddings. We consider 39 functional groups, which can be found in the supplementary materials.

We design a GNN to model molecule representations in $\mathcal{G}_x$. Guided by chemical reactions, it employs an attention mechanism for aggregating neighbor information. Specifically, for molecule m_i in reaction $x : R \to P$ (i.e., $m_i \in R \cup P$), its neighbor aggregation in the l-th layer is defined as:

$$\mathbf{m}_i^l = \mathbf{m}_i^{l-1} + \sum\nolimits_{j \in \mathcal{N}_i} \pi_{ij}^l \mathbf{m}_j^{l-1}, \tag{3}$$

where $\mathbf{m}_i^l$ is m_i's representation ($\mathbf{m}_i^0 = \mathbf{m}_i$), π_{ij} denotes the attention score between m_i and its neighbor $m_j \in \mathcal{N}_i$ in $\mathcal{G}_x$. To enhance computational efficiency, we sample a fixed-size neighbor subset from $\mathcal{N}_i$ for each molecule. The size of the subset is a predefined constant Q.

The attention score π_{ij} is defined as follows:

$$\pi_{ij}^l = \frac{\exp(\mathbf{W}_a^l(\mathbf{m}_i^{l-1} \odot \mathbf{m}_j^{l-1} \odot \mathbf{c}^{l-1}))}{\sum_{k \in \mathcal{N}_i} \exp(\mathbf{W}_a^l(\mathbf{m}_i^{l-1} \odot \mathbf{m}_k^{l-1} \odot \mathbf{c}^{l-1}))}, \tag{4}$$

where $\odot$ denotes the element-wise product between vectors, $\mathbf{W}_a$ is the weight matrix, and $\mathbf{c}$ is the contextual information for molecule m_i in chemical reaction $x : R \to P$. The reaction context $\mathbf{c}$ is defined as $\mathbf{c}^l = \sum_{m_u \in R \cup P} \mathbf{m}_u^l$. Our attention mechanism considers the reaction context, allowing for the control in neighbor message passing.

After stacking L layers of GNN, we utilize representation $\mathbf{m}_i^L$ as molecule m_i's reaction-level representation $\mathbf{h}_i^X$.

Molecule Representation Optimization

For each molecule m_i, we derive its two representations (i.e., $\mathbf{h}_i^A$ and $\mathbf{h}_i^X$) from dual graph learning. To model m_i's final representation $\mathbf{h}_i$, we use a linear transformation with weight $\mathbf{W}_c$ and concatenation operation $\|$, as $\mathbf{h}_i = \mathbf{W}_c(\mathbf{h}_i^A \| \mathbf{h}_i^X)$.

To further optimize these molecular representations, we introduce a reaction-based relation learning task and a cross-view contrastive learning task. The former models the relation between reactants and products in each reaction. The latter is designed to enhance the cooperative association between molecular and reaction-aware graph learning.

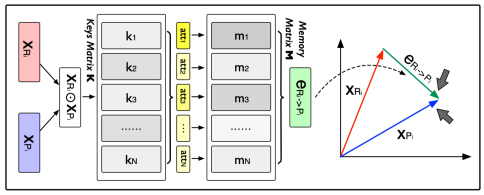


Fig. 4. Reaction-based relation learning.

Reaction-based relation learning task

This task models reactant and product pairs using the relation vector, as shown in Fig.1(b_2). Given a chemical reaction $x_i : R_i \to P_i$, we assume that equation $\mathbf{x}_{R_i} + \mathbf{e}_{R_i \to P_i} = \mathbf{x}_{P_i}$ holds, where $\mathbf{x}_{R_i}$, $\mathbf{x}_{P_i}$, and $\mathbf{e}_{R_i \to P_i}$ are the representations of reactant R_i, product P_i, and relation from R_i to P_i, respectively. First, we use the summation operation to define $\mathbf{x}_{R_i}$ and $\mathbf{x}_{P_i}$, as:

$$\mathbf{x}_{R_i} = \sum\nolimits_{m_j \in R_i} \mathbf{h}_j, \qquad \mathbf{x}_{P_i} = \sum\nolimits_{m_k \in P_i} \mathbf{h}_k. \tag{5}$$

To model the relation vector $\mathbf{e}_{R_i \to P_i}$ of reaction x_i, we introduce a memory network (as shown in Fig.4) which takes a reactant-product pair $(\mathbf{x}_{R_i}, \mathbf{x}_{P_i})$ as input, and outputs the relation vector $\mathbf{e}_{R_i \to P_i}$ of equal dimension as $\mathbf{x}_{R_i}$ and $\mathbf{x}_{P_i}$.

Let $\mathbb{R}^d$ be the dimension of $\mathbf{x}_{R_i}$ and $\mathbf{x}_{P_i}$. The network first introduces a key matrix $\mathbf{K} \in \mathbb{R}^{N \times d}$ to calculate an attention vector $\mathbf{att} \in \mathbb{R}^N$. The element $att_n \in \mathbf{att}$ is defined as:

$$att_n = \frac{\exp\left((\mathbf{x}_{R_i} \odot \mathbf{x}_{P_i})^\top \mathbf{k}_n\right)}{\sum_{m=1}^N \exp\left((\mathbf{x}_{R_i} \odot \mathbf{x}_{P_i})^\top \mathbf{k}_m\right)}, \tag{6}$$

where $\mathbf{k} \in \mathbb{R}^d$ is the vector element of matrix $\mathbf{K}$.

Next, we introduce a memory matrix $\mathbf{M} \in \mathbb{R}^{N \times d}$ to generate the relation vector $\mathbf{e}_{R_i \to P_i}$, as follows:

$$\mathbf{e}_{R_i \to P_i} = \sum\nolimits_{n=1}^N att_n \cdot \mathbf{m}_n, \tag{7}$$

where $\mathbf{m} \in \mathbb{R}^d$ is the vector element of matrix $\mathbf{M}$.

The relation vector $\mathbf{e}_{R_i \to P_i}$ is a weighted representation of $\mathbf{M}$. Intuitively, the memory matrix $\mathbf{M}$ can be interpreted as a store of conceptual building blocks that can be used to describe the relations between reactants and products.

After the relation modeling, we define the following score function for reactant-product pair R_i and P_i of reaction x_i:

$$\text{S}(R_i, P_i) = \|\mathbf{x}_{R_i} + \mathbf{e}_{R_i \to P_i} - \mathbf{x}_{P_i}\|_2. \tag{8}$$

For optimization, we utilize a contrastive learning method similar to (Radford *et al.*, 2021). In a minibatch of reaction

data $\mathcal{X}_\mathcal{B}$, we identify matched reactant-product pairs as positive pairs, aiming to minimize their embedding differences, while unmatched pairs are considered negative pairs, with an objective to maximize their embedding discrepancies. We use the margin-based loss function (Bordes *et al.*, 2013), as follows:

$$\mathcal{L}_E = \frac{1}{|\mathcal{X}_\mathcal{B}|}\sum_{x_i\in\mathcal{X}_\mathcal{B}} \mathrm{S}(R_i,P_i)+ \frac{1}{|\mathcal{X}_\mathcal{B}|^2-|\mathcal{X}_\mathcal{B}|}\sum_{x_i\in\mathcal{X}_\mathcal{B}}\sum_{x_j\in\mathcal{X}_\mathcal{B}}\max\Big(\gamma-\mathrm{S}(R_i,P_j),0\Big), \tag{9}$$

where $x_i \neq x_j$ and $\gamma > 0$ is a margin hyperparameter.

Cross-view contrastive learning task

This task enhances representations of molecules by aligning outputs from molecular and reaction-aware graph learning modules. We consider different views of the same reactant or product as positive pairs, while views of different reactants or products form negative pairs. This method enables our model to learn distinct representations by contrasting these positive and negative instances. Since contrastive tasks of reactants and products are similar, we illustrate the reactant side task.

Specifically, for a reaction x_i, we first generate its two reactant representations $\mathbf{x}^A_{R_i}$ and $\mathbf{x}^X_{R_i}$ from the molecular and reaction-aware graph learning modules, respectively, as:

$$\mathbf{x}^A_{R_i} = \sum\nolimits_{m_j\in R_i}\mathbf{h}^A_j, \qquad \mathbf{x}^X_{R_i} = \sum\nolimits_{m_k\in R_i}\mathbf{h}^X_k. \tag{10}$$

Then, we enforce the separation of different reactant representations and align those that are identical. To achieve this, we employ InfoNCE (Oord *et al.*, 2018) to define the cross-view contrastive loss for reactants as follows:

$$\mathcal{L}_{C_r} = \sum_{x_i\in\mathcal{X}_\mathcal{B}} -\log\frac{\exp\left(c(\mathbf{x}^A_{R_i},\mathbf{x}^X_{R_i})/\tau\right)}{\sum_{x_j\in\mathcal{X}_\mathcal{B}}\exp\left(c(\mathbf{x}^A_{R_i},\mathbf{x}^X_{R_j})/\tau\right)}, \tag{11}$$

where $\mathcal{X}_\mathcal{B}$ denotes a minibatch of reaction data, $c(\cdot,\cdot)$ is the cosine similarity, and τ is a hyper-parameter for the softmax function. Here we use a minibatch-based contrastive strategy for its efficiency in terms of both time and memory.

Similarly, we calculate contrastive loss $\mathcal{L}_{C_p}$ for the product side. Combining two losses, we formulate the objective function of our cross-view contrastive task as $\mathcal{L}_C = \mathcal{L}_{C_r} + \mathcal{L}_{C_p}$.

Optimization Function

Our RXGL is trained using a weighted sum of losses from the reaction relation learning and cross-view contrastive tasks.

$$\mathcal{L} = \mathcal{L}_E + \alpha\mathcal{L}_C + \lambda||\Theta||^2_\mathrm{F}, \tag{12}$$

where Θ is model parameters, and α and λ adjust the cross-view contrastive task and L2-regularization, respectively.

We use the minibatch-based negative sampling strategy for two tasks, offering advantages over the traditional approach (Mikolov *et al.*, 2013). First, it requires no extra memory to store negative samples. Second, negative samples are refreshed each epoch due to the shuffling of training instances, saving the time for manual re-sampling.

Experiments

In this section, we conduct extensive experiments to show the effectiveness of our approach RXGL in several downstream tasks, including product prediction, reaction classification, and molecular property prediction.

Product Prediction

Reaction product prediction is a fundamental problem in the biology and chemistry field, which aims to predict the products of a reaction based on the reactants.

Dataset. We conduct experiments on two biochemistry benchmark datasets: USPTO-15K (Jin *et al.*, 2017), containing 15,000 reactions, and USPTO-50K (Liu *et al.*, 2024), comprising 50,000 reactions. Each dataset was divided into training, validation, and test sets using an 8:1:1 ratio.

Evaluation Protocol. Following (Wang *et al.*, 2022a), we formulate the product prediction as a ranking task. Let $\mathcal{X}_{test}$ and P_{test} be the sets of reactions and products in the test set. For each reaction $x_i : R_i \rightarrow P_i$ in $\mathcal{X}_{test}$, we rank all products $P_j \in P_{test}$ based on the score function $S(R_i, P_j)$ (i.e., Eq.(8)). Then, the ranking of the ground-truth candidate can be used for the evaluation. Our evaluation metrics are MRR (mean reciprocal rank) and Hit@1 (hit ratio at a cut-off value of 1). We conduct each experiment five times, reporting both the mean and standard deviation on the test set, with the results selected based on the best MRR in the validation set.

Baselines. Our RXGL is compared with Mol2vec (Jaeger *et al.*, 2018), MolBERT (Fabian *et al.*, 2020), and MolR (Wang *et al.*, 2022a). For Mol2vec and MolBERT, we employ their pre-trained models to generate embeddings for reactants and products. Then, aligning with MolR, the scoring function is defined through an inner product. For MolR and our RXGL, we use four GNNs (i.e., GCN, GAT, SAGE, and TAG) as encoders for the molecular graph. We present the details of these encoders in the supplementary materials.

Hyperparameter Settings. We train our model RXGL for 50 epochs with a batch size of 1,024, using Adam optimizer with a learning rate of 10^{-4}. The number of GNN layers in molecular and reaction-aware graph learning is set to 2 and 1, respectively, and the output dimension of all layers is 256. In addition, neighbor sampling size Q, memory slice size N, margin γ, temperature τ, and balancing factor α are set to 10, 10, 4, 0.1, and 10^{-3}, respectively. The result of the key hyperparameter sensitivity is reported in Section 4.4.

Performance Comparison. The results are shown in Table 1. We find that our RXGL performs best. To be specific, RXGL-SAGE achieves 14.1% average MRR gain and 16.3% average Hit@1 gain over baselines on the USPTO-15K dataset, showing the effectiveness of RXGL in the product prediction.

Table 1. Results of the product prediction task on the two datasets. The numbers in brackets are the standard deviations.

Methods	USPTO-15K		USPTO-50K	
	MRR	Hit@1	MRR	Hit@1
Mol2vec	0.519	0.468	0.835	0.801
MolBERT	0.790	0.734	0.913	0.874
MolR-GCN	$0.883_{(0.005)}$	$0.847_{(0.007)}$	$0.958_{(0.003)}$	$0.944_{(0.006)}$
MolR-GAT	$0.881_{(0.003)}$	$0.846_{(0.002)}$	$0.952_{(0.002)}$	$0.931_{(0.004)}$
MolR-SAGE	$0.932_{(0.006)}$	$0.905_{(0.003)}$	$0.972_{(0.005)}$	$0.960_{(0.005)}$
MolR-TAG	$0.925_{(0.005)}$	$0.898_{(0.004)}$	$0.974_{(0.010)}$	$0.965_{(0.009)}$
RXGL-GCN	$0.927_{(0.003)}$	$0.899_{(0.005)}$	$0.965_{(0.007)}$	$0.954_{(0.002)}$
RXGL-GAT	$0.925_{(0.007)}$	$0.894_{(0.006)}$	$0.967_{(0.009)}$	$0.958_{(0.011)}$
RXGL-SAGE	$\mathbf{0.956}_{(0.004)}$	$\mathbf{0.936}_{(0.007)}$	$\mathbf{0.982}_{(0.005)}$	$0.973_{(0.003)}$
RXGL-TAG	$0.941_{(0.006)}$	$0.919_{(0.009)}$	$0.979_{(0.004)}$	$\mathbf{0.974}_{(0.009)}$

Reaction Classification

Reaction classification aims to predict the class of reactions.

Dataset. Our RXGL is evaluated using two datasets, i.e., Schneider and USPTO-MTL. Specifically, Schneider (Schneider *et al.*, 2015) comprises 38,800 reactions across 46 classes,

and USPTO-MTL (Lu and Zhang, 2022) includes 143,535 reactions across 1,000 classes. Each dataset is randomly split into training, validation, and test sets in an 8:1:1 ratio.

Baselines and Experiment Settings. Similar to product prediction, we compare our RXGL with Mol2vec, MolBERT, and MolR. We employ our model pre-trained on the USPTO-50k dataset to process all datasets, which includes generating embeddings for reactants and products in each reaction and concatenating them to create a unified reaction feature. For prediction, we use an MLP as the decoder. Accuracy and Recall are used as evaluation metrics. For each experiment, we perform five times and report the mean and standard deviation of the results on the test set.

Performance Comparison. Table 2 shows the results of the reaction classification task. We find that our RXGL outperforms baselines. For example, RXGL-SAGE achieves an average Accuracy increase of 4.0% on Schneider dataset. These results highlight RXGL's efficacy in the reaction classification task, and suggest that the molecule representations it learns are effectively transferable to the downstream task.

Table 2. Results of the reaction classification task. The numbers in brackets are the standard deviations.

Methods	Schneider		USPTO-MTL	
	Accuracy	Recall	Accuracy	Recall
Mol2vec	$0.856_{(0.012)}$	$0.850_{(0.009)}$	$0.751_{(0.016)}$	$0.629_{(0.014)}$
MolBERT	$0.849_{(0.014)}$	$0.847_{(0.012)}$	$0.738_{(0.009)}$	$0.583_{(0.008)}$
MolR-GCN	$0.879_{(0.013)}$	$0.882_{(0.013)}$	$0.853_{(0.020)}$	$0.813_{(0.015)}$
MolR-GAT	$0.870_{(0.024)}$	$0.873_{(0.021)}$	$0.862_{(0.017)}$	$0.834_{(0.014)}$
MolR-SAGE	$0.882_{(0.030)}$	$0.881_{(0.027)}$	$0.874_{(0.023)}$	$0.839_{(0.026)}$
MolR-TAG	$0.891_{(0.025)}$	$0.895_{(0.026)}$	$0.888_{(0.024)}$	$0.852_{(0.024)}$
RXGL-GCN	$0.887_{(0.016)}$	$0.889_{(0.017)}$	$0.875_{(0.018)}$	$0.849_{(0.015)}$
RXGL-GAT	$0.872_{(0.026)}$	$0.874_{(0.020)}$	$0.873_{(0.019)}$	$0.836_{(0.016)}$
RXGL-SAGE	$\mathbf{0.907}_{(0.029)}$	$\mathbf{0.906}_{(0.025)}$	$0.889_{(0.014)}$	$0.858_{(0.018)}$
RXGL-TAG	$0.899_{(0.033)}$	$0.901_{(0.031)}$	$\mathbf{0.895}_{(0.019)}$	$\mathbf{0.867}_{(0.018)}$

Molecular Property Prediction

This task is to predict labels of given molecules, which is a classical task to evaluate learned molecule representations.

Dataset. We evaluate our RXGL on four datasets (Wu *et al.*, 2018): BBBP, BACE, Tox21, and ClinTox. Each dataset contains molecule SMILES as well as labels indicating the property. Readers can refer to (Wu *et al.*, 2018) for a detailed introduction to these four public datasets.

Baselines. We compare our RXGL (i.e., variant RXGL-GCN) with the following four class methods: (1) SMILES-based methods: ChemBERTa (Chithrananda *et al.*, 2020) and MolBERT (Fabian *et al.*, 2020); (2) Fingerprint-based methods: Mol2vec (Jaeger *et al.*, 2018), ECFP4 (Rogers and Hahn, 2010), GraphConv (Duvenaud *et al.*, 2015), Weave (Kearnes *et al.*, 2016), D-MPNN (Yang *et al.*, 2019), CDDD (Winter *et al.*, 2019); (3) GNN-based methods: GraphCL (You *et al.*, 2020), GraphLoG (Xu *et al.*, 2021), EdgePred (Hamilton *et al.*, 2017), AttrMask (Hu *et al.*, 2019), GPT-GNN (Hu *et al.*, 2020), InfoGraph (Sun *et al.*, 2020), ContextPred (Hu *et al.*, 2019), G-Motif (Rong *et al.*, 2020), JOAO (You *et al.*, 2021), and GraphMVP (Liu *et al.*, 2022); (4) Reaction-enhanced methods: MolR (Wang *et al.*, 2022a) and ReaKE (Wang *et al.*, 2022b).

Experiment Settings. We split all datasets into training, validation, and test sets in an 8:1:1 ratio, using two split types: random and scaffold. Our model pre-trained on USPTO-50K datasets is used to generate molecule embeddings, which are formed by concatenating outputs from the molecular graph learning module with the sum of molecule functional group embeddings. Then, we input the molecule embeddings along with their labels into a logistic regression model. We use the AUC (area under the curve) as our evaluation metric. Each experiment is conducted five times, and we report the mean and standard deviation of the results on the test set.

Table 3. Molecular property prediction results (split type: *random split*). The numbers in brackets are the standard deviations. The results of symbols ★ and ♠ are taken from MolR and ReaKE.

Methods	BBBP	BACE	Tox21	ClinTox
ChemBERTa★	0.643	–	0.728	0.733
MolBERT★	$0.762_{(0.000)}$	$0.866_{(0.000)}$	–	–
Mol2vec★	$0.872_{(0.021)}$	$0.862_{(0.027)}$	$0.803_{(0.041)}$	$0.841_{(0.062)}$
ECFP4★	0.729	0.867	0.822	0.799
GraphConv★	0.690	0.783	0.829	0.807
Weave★	0.671	0.806	0.820	0.832
D-MPNN★	0.708	–	0.688	0.906
CDDD★	$0.761_{(0.000)}$	$0.833_{(0.000)}$	–	–
GraphCL★	$0.695_{(0.005)}$	$0.782_{(0.012)}$	$0.754_{(0.009)}$	$0.701_{(0.019)}$
GraphLoG★	$0.725_{(0.008)}$	$0.835_{(0.012)}$	$0.757_{(0.005)}$	$0.767_{(0.033)}$
MolR★	$0.895_{(0.031)}$	$0.875_{(0.023)}$	$0.820_{(0.028)}$	$0.913_{(0.043)}$
ReaKE♠	–	0.898	0.824	0.874
RXGL	$\mathbf{0.901}_{(0.024)}$	$\mathbf{0.906}_{(0.017)}$	$\mathbf{0.852}_{(0.026)}$	$\mathbf{0.921}_{(0.030)}$

Table 4. Property prediction results (split type: *scaffold split*). The numbers in brackets are the standard deviations. The results of symbols ★ and ♠ are taken from GraphMVP and ReaKE.

Methods	BBBP	BACE	Tox21	ClinTox
EdgePred★	$0.645_{(0.031)}$	$0.646_{(0.047)}$	$0.745_{(0.004)}$	$0.558_{(0.062)}$
AttrMask★	$0.702_{(0.005)}$	$0.772_{(0.014)}$	$0.742_{(0.008)}$	$0.686_{(0.096)}$
GPT-GNN★	$0.645_{(0.011)}$	$0.776_{(0.005)}$	$0.753_{(0.005)}$	$0.578_{(0.031)}$
InfoGraph★	$0.692_{(0.008)}$	$0.739_{(0.025)}$	$0.730_{(0.007)}$	$0.751_{(0.050)}$
ContextPred★	$0.712_{(0.009)}$	$0.786_{(0.014)}$	$0.733_{(0.005)}$	$0.737_{(0.040)}$
G-Contextual★	$0.703_{(0.016)}$	$0.792_{(0.003)}$	$0.752_{(0.003)}$	$0.599_{(0.082)}$
G-Motif★	$0.664_{(0.034)}$	$0.734_{(0.040)}$	$0.732_{(0.008)}$	$0.778_{(0.020)}$
JOAO★	$0.660_{(0.006)}$	$0.729_{(0.020)}$	$0.744_{(0.007)}$	$0.663_{(0.039)}$
GraphMVP★	$0.724_{(0.016)}$	$0.812_{(0.009)}$	$\mathbf{0.744}_{(0.002)}$	$0.775_{(0.042)}$
MolR♠	–	0.774	0.670	0.830
ReaKE♠	–	0.781	0.713	0.862
RXGL	$\mathbf{0.729}_{(0.015)}$	$\mathbf{0.825}_{(0.006)}$	$0.736_{(0.003)}$	$\mathbf{0.912}_{(0.011)}$

Performance Comparison. Different baselines leverage different strategies (i.e., random or scaffold) to split datasets. We compare our model with baselines by adopting their selected dataset splitting. The AUC results for molecular property prediction are presented in Tables 3 and 4. The baseline results are taken from the literature. Our RXGL shows strong performance across four datasets, e.g., RXGL exhibits average AUC gains of 17.8%, 7.2%, 7.9%, and 11.3% on four datasets under random splitting, showing that learned molecule representations effectively transfer to molecule-related tasks.

Hyper-Parameter Study

We study three hyper-parameters on RXGL-GCN: embedding size (d), balancing factor (α), and neighbor sampling size (Q), as shown in Fig.5. We find: (1) A larger d typically enhances performance by encoding more information, yet an excessively large d increases the parameter size. We suggest setting d at 256 to balance performance and storage requirements. (2) Nice performance is achieved when α is set to 0.001. We hypothesize that a small α might inadequately reinforce the cooperative associations between molecular and reaction-aware graph learning. (3) The rank accuracy improves with increasing Q but eventually decreases. This decline may be due to the introduction of irrelevant information from too many neighbors.

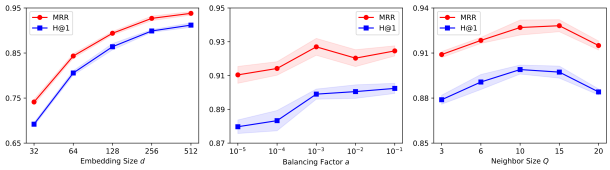


Fig. 5. Hyper-parameter study on the USPTO-15K dataset.

Embedding Analysis

This subsection studies the learned molecule embeddings and reaction-based relation embeddings (i.e., Eq.(7)).

Molecule Embedding Analysis

We first use RXGL-GCN to generate embeddings of molecules in the BBBP dataset and visualize them utilizing the t-SNE method (Van der Maaten and Hinton, 2008). Specifically, we select the molecules' permeability property, molecule size (i.e., the number of atoms), and 39 functional group properties. These 39 functional groups are provided by RDKit (Landrum, 2013). Due to the space limitation, we present visualizations in the main text for the permeability, the molecule size, and the hydroxyl functional group properties. More detailed information about these functional groups and remaining experimental results can be found in supplementary materials. From visualization results, we find that: Fig.6(a) illustrates molecules colored according to their permeability properties. Several distinct clusters of non-permeable molecules are observed. Fig.6(b) shows molecules colored by size. The embedding space distinctly segregates small molecules (located in the right region) from large molecules (located in the left region). Fig.6(c) shows molecules colored based on the number of the hydroxyl functional groups (i.e., '-OH'). Molecules with a higher number of hydroxyl groups are mainly on the left side of the embedding space, while those without this group are primarily found on the right side. In summary, these results indicate that the molecular embeddings generated by our RXGL framework exhibit a certain degree of correlation with the aforementioned three molecular properties.

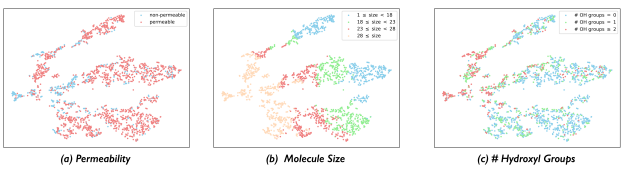


Fig. 6. Visualized molecule embeddings on the BBBP dataset.

Reaction-based Relation Embedding Analysis

In our RXGL, we learn relation embeddings between reactants and products in chemical reactions. This subsection examines the meaningfulness of these relation embeddings.

A key feature of chemical reactions is the change of reactants into products. For example, this process involves breaking old bonds and forming new ones, resulting in a change in the number of bonds before and after the reaction. Intuitively, reactions with similar changes yield similar relation embeddings. To analyze this, we randomly select five reactions (labeled a, b, c, d, and e) from the test set of USPTO-15K dataset and compute the cosine similarity of their relation embeddings, as shown in the left subfigure of Fig.7. Take reaction a as an example, we observe a high similarity between a and c and a low similarity between a and d. We characterize chemical reaction changes by the difference in bond number (N_B) and ring number (N_R) in reactants and products. The middle and right subfigures of Fig.7 show differences between these five chemical reactions in terms of bond number changes and ring number changes. For example, a small square in the middle subfigure represents $|N_B(i) - N_B(j)|$, which means the differences between reactions i and j in terms of bond number changes. From the results, we find that the high cosine similarity between reactions a and c is likely due to their similar bond and ring number changes. Conversely, the low similarity between reactions a and d can be attributed to their significant differences in bond and ring number changes.

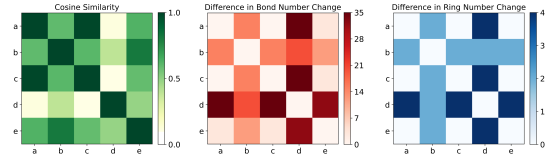


Fig. 7. Examples of relation embedding comparison.

We further conduct the macro-analysis. We calculate the cosine similarity between every reaction pair in the test set and examine the differences in bond number and ring number changes between each pair. We show the results on the USPTO-15K dataset in Fig.8, where the y-axis represents the average change within a certain similarity interval. We find that reaction pairs with high cosine similarities tend to exhibit small differences in bond and ring number changes, and vice versa. The above observation underscores the significance of our learned relation embeddings, suggesting their capability to model the changes occurring in chemical reactions.

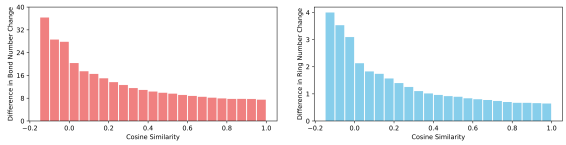


Fig. 8. Macro-analysis on the USPTO-15K dataset.

Conclusion and Future Work

This paper proposes RXGL that uses reactions as domain knowledge in MRL. RXGL integrates molecular and reaction-aware graph learning modules to model molecule representations. Also, we enhance molecule representations using a reaction-based relation learning task and a cross-view contrastive task. Experiment results show that RXGL achieves strong performance across a range of downstream tasks. We believe that the insights of this study will provide valuable guidance for future research exploring the utilization of reactions in MRL. For future work, we plan to consider the incorporation of stereochemical information into RXGL, which will enhance the accuracy and applicability of molecule representations.

Competing interests

No competing interest is declared.

Funding

The authors thank the anonymous reviewers for their valuable suggestions. This research has in part been funded by Research Council of Finland (grants 339421 and 345802) as well as Jane and Aatos Erkko Foundation (BIODESIGN project). Elena Casiraghi acknowledges travel support from the European Union's Horizon 2020 research and innovation programme

under grant agreement No 951847. This paper is supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence).

References

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *NIPS*, **26**.

Chithrananda, S., Grand, G., and Ramsundar, B. (2020). Chemberta: Large-scale self-supervised pretraining for molecular property prediction.

Du, J., Zhang, S., Wu, G., Moura, J. M., and Kar, S. (2017). Topology adaptive graph convolutional networks.

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In *NIPS*, pages 2224–2232.

Fabian, B., Edlich, T., Gaspar, H., Segler, M., Meyers, J., Fiscato, M., and Ahmed, M. (2020). Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *NIPS*, **30**.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. (2019). Strategies for pre-training graph neural networks. *ICLR*.

Hu, Z., Dong, Y., Wang, K., Chang, K.-W., and Sun, Y. (2020). Gpt-gnn: Generative pre-training of graph neural networks. In *KDD*, pages 1857–1867.

Jaeger, S., Fulle, S., and Turk, S. (2018). Mol2vec: unsupervised machine learning approach with chemical intuition. *JCIM*, **58**(1), 27–35.

Jin, W., Coley, C., Barzilay, R., and Jaakkola, T. (2017). Predicting organic reaction outcomes with weisfeiler-lehman network. *NIPS*, **30**.

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. (2016). Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, **30**(8), 595–608.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Landrum, G. (2013). Rdkit documentation. *Release*, **1**, 4.

Li, A., Yang, B., Huo, H., and Hussain, F. (2022). Hypercomplex graph collaborative filtering. In *WWW*, pages 1914–1922.

Li, A., Yang, B., Huo, H., Hussain, F. K., and Xu, G. (2024). Structure-and logic-aware heterogeneous graph learning for recommendation. In *ICDE*, pages 544–556. IEEE.

Liu, J., Yan, C., Yu, Y., Lu, C., Huang, J., Ou-Yang, L., and Zhao, P. (2024). Mars: a motif-based autoregressive model for retrosynthesis prediction. *Bioinformatics*, **40**.

Liu, S., Wang, H., Lasenby, J., and Tang, J. (2022). Pre-training molecular graph representation with 3d geometry.

Lu, J. and Zhang, Y. (2022). Unified model for multitask reaction predictions with explanation. *JCIM*, **62**(6), 1376–1387.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *NIPS*, **26**.

Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., and Weston, J. (2016). Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.* (2021). Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *JCIM*, **50**(5), 742–754.

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. *NIPS*, **33**, 12559–12571.

Schneider, N., Lowe, D. M., Sayle, R. A., and Landrum, G. A. (2015). Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *JCIM*, **55**(1), 39–53.

Sun, F.-Y., Hoffmann, J., Verma, V., and Tang, J. (2020). Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *ICLR*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, **9**(11).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2018). Graph attention networks. In *ICLR*.

Wagner, A. and Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings of the Royal Society of London*, **268**(1478), 1803–1810.

Wang, H., Li, W., Jin, X., Cho, K., Ji, H., Han, J., and Burke, M. D. (2022a). Chemical-reaction-aware molecule representation learning. In *ICLR*.

Wang, Y., Zheng, S., Rao, J., Luo, Y., and Yang, Y. (2022b). Reake: Contrastive molecular representation learning with chemical synthetic knowledge graph.

Wen, M., Spotte-Smith, E. W. C., Blau, S. M., McDermott, M. J., Krishnapriyan, A. S., and Persson, K. A. (2023). Chemical reaction networks and opportunities for machine learning. *Nature Computational Science*, **3**(1), 12–24.

Winter, R., Montanari, F., Noé, F., and Clevert, D.-A. (2019). Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical science*, **10**(6), 1692–1701.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. (2018). Moleculenet: a benchmark for molecular machine learning. *Chemical science*, **9**(2), 513–530.

Xu, M., Wang, H., Ni, B., Guo, H., and Tang, J. (2021). Self-supervised graph-level representation learning with local and global structure. In *ICML*, pages 11548–11558. PMLR.

Yang, K., Swanson, K., Jin, W., Coley, C., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B. P., Palmer, A., Settels, V., *et al.* (2019). Are learned molecular representations ready for prime time? *arXiv preprint arXiv:1904.01561,*.

Yi, H.-C., You, Z.-H., Huang, D.-S., and Kwoh, C. K. (2022). Graph representation learning in bioinformatics: trends, methods and applications. *BIB*, **23**(1), bbab340.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. (2020). Graph contrastive learning with augmentations. *NIPS*, **33**, 5812–5823.

You, Y., Chen, T., Shen, Y., and Wang, Z. (2021). Graph contrastive learning automated. In *ICML*.

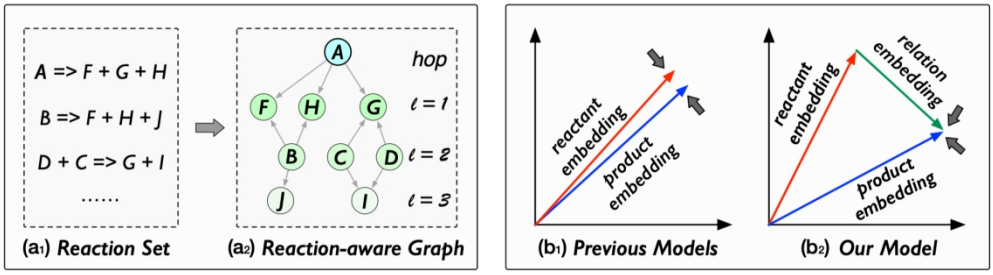


Fig. 1

407x109mm (118 x 118 DPI)

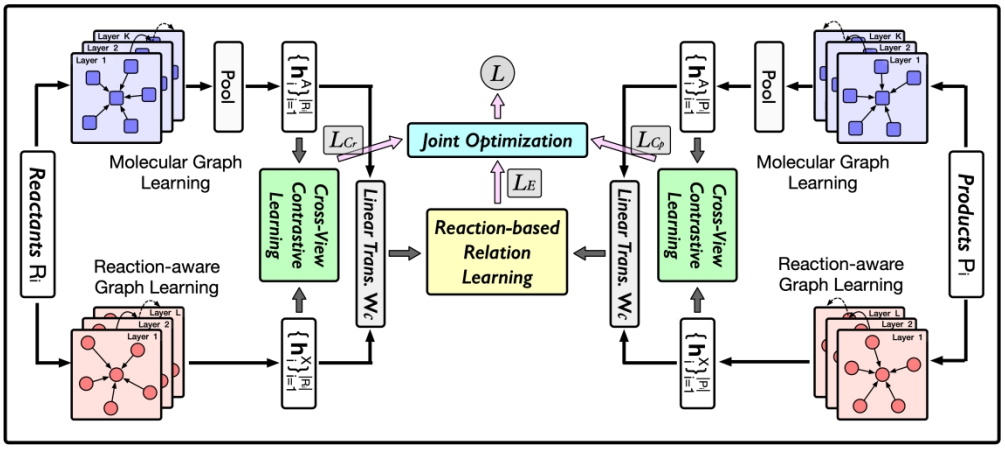
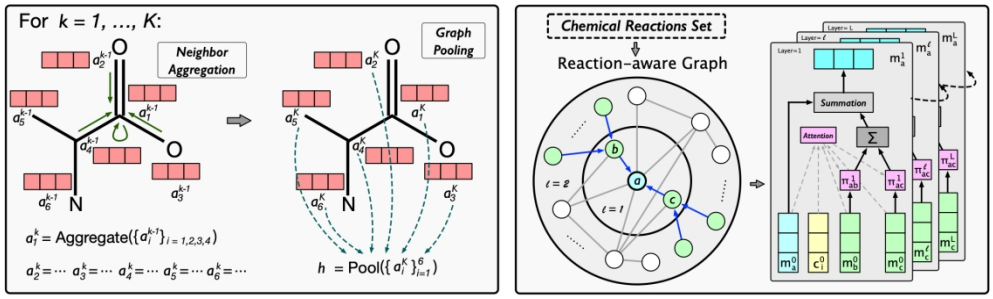


Fig. 2

799x353mm (118 x 118 DPI)



(a) Molecular Graph Learning (on Alanine $C_3H_7NO_2$) *(b) Reaction-aware Graph Learning*

Fig. 3

454x147mm (118 x 118 DPI)

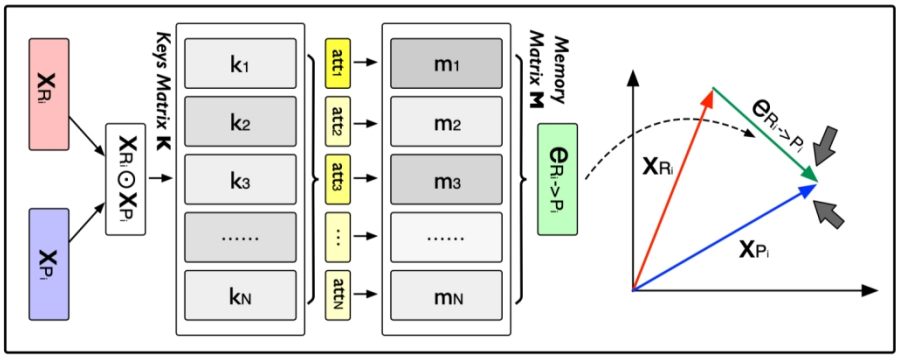


Fig. 4

363x127mm (118 x 118 DPI)

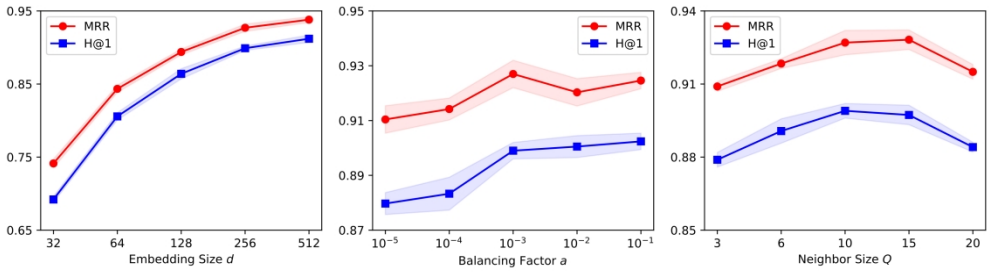
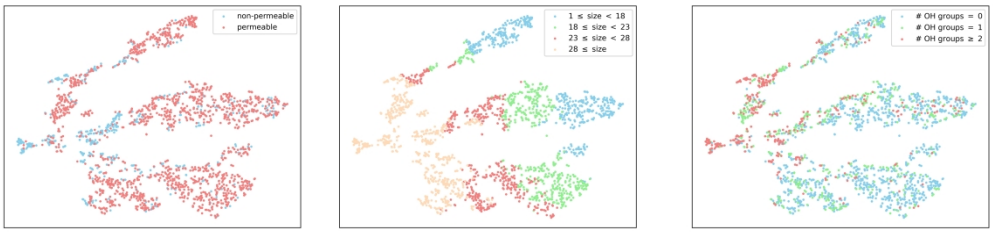


Fig. 5

768x214mm (118 x 118 DPI)



(a) Permeability *(b) Molecule Size* *(c) # Hydroxyl Groups*

Fig. 6

1294x334mm (118 x 118 DPI)

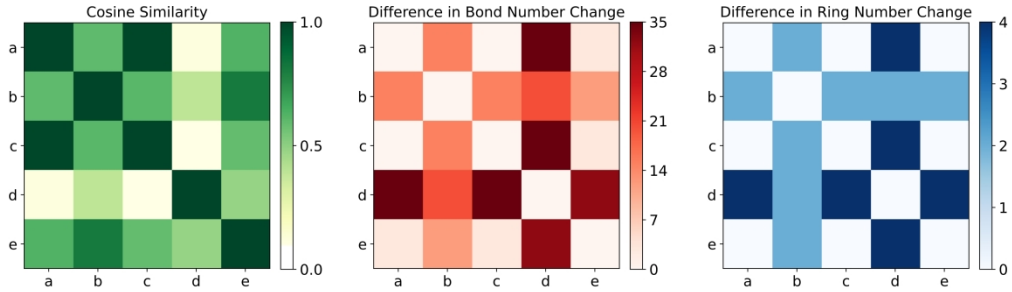


Fig. 7

930x263mm (118 x 118 DPI)

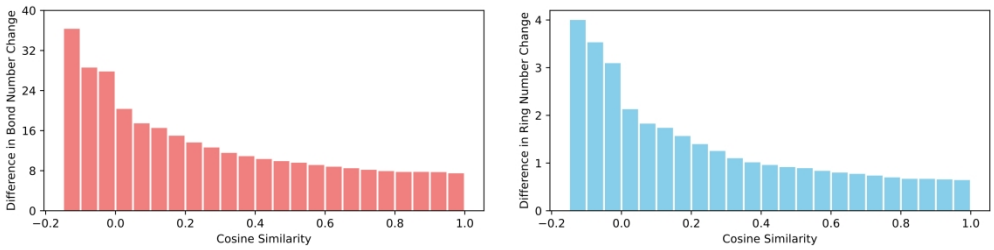


Fig. 8

793x202mm (118 x 118 DPI)