

Molecular Graph Representation Learning via Structural Similarity Information

Chengyu Yao^{1,2*}, Hong Huang^{1,3*}, Hang Gao^{1,2}, Fengge Wu^{1,2} (✉), Haiming Chen³, and Junsuo Zhao^{1,2}

¹ University of Chinese Academy of Sciences, 100081 Beijing, China

² National Key Laboratory of Space Integrated Information System, Institute of Software Chinese Academy of Sciences, 100081 Beijing, China

³ Key Laboratory of System Software (Chinese Academy of Sciences) and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, 100190 Beijing, China

{huanghong, chm}@ios.ac.cn

{yaochengyu2023, gaohang, fengge, junsuo}@iscas.ac.cn

Abstract. Graph Neural Networks (GNNs) have been widely employed for feature representation learning in molecular graphs. Therefore, it is crucial to enhance the expressiveness of feature representation to ensure the effectiveness of GNNs. However, a significant portion of current research primarily focuses on the structural features within individual molecules, often overlooking the structural similarity between molecules, which is a crucial aspect encapsulating rich information on the relationship between molecular properties and structural characteristics. Thus, these approaches fail to capture the rich semantic information at the molecular structure level. To bridge this gap, we introduce the **Molecular Structural Similarity Motif GNN (MSSM-GNN)**, a novel molecular graph representation learning method that can capture structural similarity information among molecules from a global perspective. In particular, we propose a specially designed graph that leverages graph kernel algorithms to represent the similarity between molecules quantitatively. Subsequently, we employ GNNs to learn feature representations from molecular graphs, aiming to enhance the accuracy of property prediction by incorporating additional molecular representation information. Finally, through a series of experiments conducted on both small-scale and large-scale molecular datasets, we demonstrate that our model consistently outperforms eleven state-of-the-art baselines. The codes are available at <https://github.com/yaoyao-yaoyao-cell/MSSM-GNN>.

Keywords: Molecular property prediction · Graph neural networks · Graph representation learning · Graph kernel.

1 Introduction

Molecular Representation Learning, a critical discipline in bioinformatics and computational chemistry, has witnessed significant advancements in recent years

* These authors contributed equally to this work

[11, 22, 33]. Accurate prediction of molecular properties and activities is essential for drug discovery [17], toxicity assessment [46], and other biochemical applications [30]. Nowadays, molecular representation learning has been widely integrated with Graph Neural Networks (GNNs), which are powerful tools for processing graph data and have been successfully applied in the molecular domain [4, 15, 45]. However, most existing GNNs use the basic molecular graphs topology to obtain structural information through neighborhood feature aggregation and pooling methods [12, 21, 23]. This leads them to overlook the comprehensive chemical semantics.

To address this challenge, several emerging approaches have been proposed around molecular graphs. Specifically, some approaches [37, 45] take only the atom-level or motif-level information in heterogeneous molecular graphs as GNNs’ input to recognize common subgraphs with special meanings. By identifying the significance of ring compounds in molecular structures, Zhu et al. [48] propose the Ring-Enhanced Graph Neural Network (\mathcal{O} -GNN). Alternatively, other methods [13, 18, 43] represent the molecular using motif-aware models that consider properties of domain-specific motifs. Furthermore, there exists a multitude of techniques [1, 25, 44] that center their focus on studying the relations among substructures to recognize critical patterns hidden in motifs and improve the reliability of molecular property prediction.

Despite the considerable progress compared to traditional GNNs, most recent studies focus only on the message passing within individual molecules. The relationships between molecular structures are often ignored, which may result in the partial loss of semantic information. Moreover, the functions and properties of chemical molecules largely depend on their structures [42]. For instance, consider examples illustrated in Fig. 1. Molecules with similar structures often have similar properties. Therefore, we need to take specific measures to represent the structural similarity between molecules, which can benefit downstream tasks such as molecular property prediction.

Based on the above-mentioned considerations, we design a **Molecular Structural Similarity Motif** (MSSM) graph that empowers GNNs to capture the rich structural and semantic information from inter-molecule. The design starts by constructing a nested motif dictionary to re-represent molecular graphs. In light of the diverse node types present in motif-based molecular graphs, we propose a **Mahalanobis Weisfeiler-Lehman Shortest-Path** (MWLSP) graph kernel. This kernel is designed to assess structural similarity from both the perspectives of length and position. It overcomes the limitation of the shortest path graph kernel [2], which only retains connectivity information. By leveraging label information from different nodes and their neighbors, it provides a more granular representation of the graph, enhancing its expressiveness.

In this work, we propose a method that effectively considers inter-molecule structural similarity from a global perspective without sacrificing information in individual molecules. The method consists of the following major components: Firstly, it extracts motifs from molecules to create the motif dictionary and represents each molecule by utilizing the dictionary. Secondly, it uses our proposed

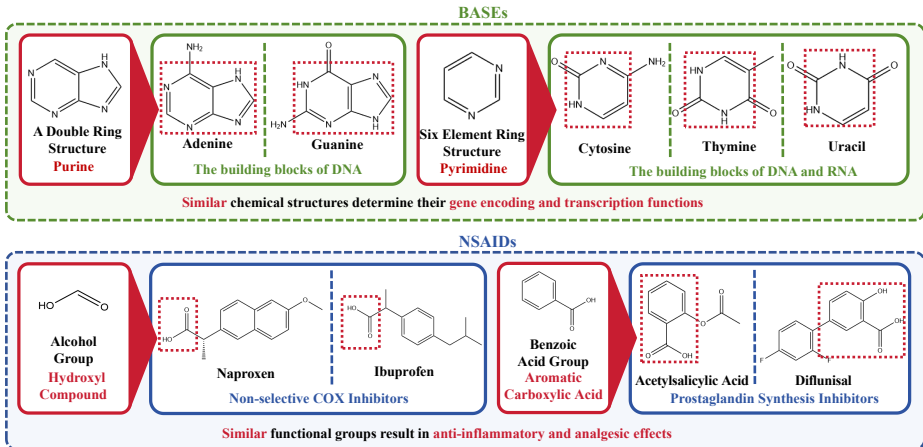


Fig. 1: Examples of molecules with similar structures often exhibit similar properties, a phenomenon observed in biological and chemical domains.

Molecular Structural Similarity Motif (MSSM) graph to exploit rich semantic information from graph motifs. Finally, it applies GNNs to learn compositional and structural feature representation for each molecular graph and their similarities based on the MSSM graph. The experimental results show that our model can significantly outperform other state-of-the-art GNN models on various molecular property prediction datasets.

To summarize, our contributions are as follows:

- Considering the actual molecule structure, we designed a novel molecular graph representation method to represent motif structural information.
- To improve the accuracy of GNNs in molecular property prediction tasks, we design a MSSM graph by employing the MWLSP graph kernel. It quantifies the similarity between molecules through graph kernel scores and obtains a more comprehensive semantic representation at the structural level.
- We show in the experiments that our model empirically outperforms state-of-the-art baselines on several benchmarks of real-world molecule datasets.

2 Related Work

2.1 Motifs in Molecular Graphs

Motif refers to the basic structure that constitutes any characteristic sequence. It can be viewed as a subgraph with a specific meaning in the molecular graph. For example, the edges in a molecular graph represent chemical bonds, and the rings represent the molecular ring structure. Several algorithms have been introduced to leverage motifs in different applications, including contrastive learning [32], self-supervised pretraining [47], generation [19] and drug-drug interaction prediction [17]. The motif extraction techniques used in the above methods, whether

relying on exact counting [5] or sampling and statistical estimation [36], have not utilized the structural similarities among motifs to enhance the expressiveness of molecular graphs.

2.2 Molecular Graph Representation Learning

DL has been widely applied to predict molecular properties. Molecules are usually represented as 1D sequences, including amino acid sequences, SMILES [41] and 2D graphs [11]. Wu et al. [38] proposed a new molecular joint representation learning framework, MMSG, based on multi-modal molecular information (from SMILES and graphs). However, these approaches cannot capture the rich information in subgraphs or graph motifs. A few works based on GNNs have been reported to leverage motif-level information. Specifically, some approaches [37, 45, 48] introduced the molecular graph representation learning method by constructing heterogeneous motif graphs from extracting different types of motifs. Alternatively, other methods [26, 48] decomposed each training molecule into fragments by breaking bonds and rings in compounds to design novel GNN variants. Although these methods obtain more expressive molecular graphs, the challenge in motif-based approaches mainly comes from the difficulty in efficiently measuring similarities between input graphs. While existing graph kernel methods [2, 7, 14, 31] can calculate scores by comparing different substructures of graphs to complete the measurement, there is currently no comparison method for motif-based molecular graphs. Therefore, our method focuses on learning motif structural information in the representation.

3 Methods

In this section, we propose a novel method to construct a Molecular Structural Similarity Motif Graph Neural Network (MSSM-GNN) (Illustrated in Fig. 2) which takes the MSSM graph as input.

Generally, the framework of the method consists of three parts: (i) Molecular graph representation; (ii) MSSM graph construction based on graph kernel; (iii) MSSM-GNN construction. Below, we explain in more detail about these parts.

3.1 Molecular Graph Representation

In molecular graphs, motifs are subgraphs that appear repeatedly and are statistically significant. Therefore, we propose a novel molecular graph representation method based on chemical domain knowledge and BRICS algorithm to represent molecular structural information better. It considers both the internal atomic structure and the overall impact of special functional groups in the molecule. Its main process consists of the following two steps: (i) Motif Dictionary; (ii) Molecular Graph Re-representation.

breaking retrosynthetically interesting chemical substructures [10]

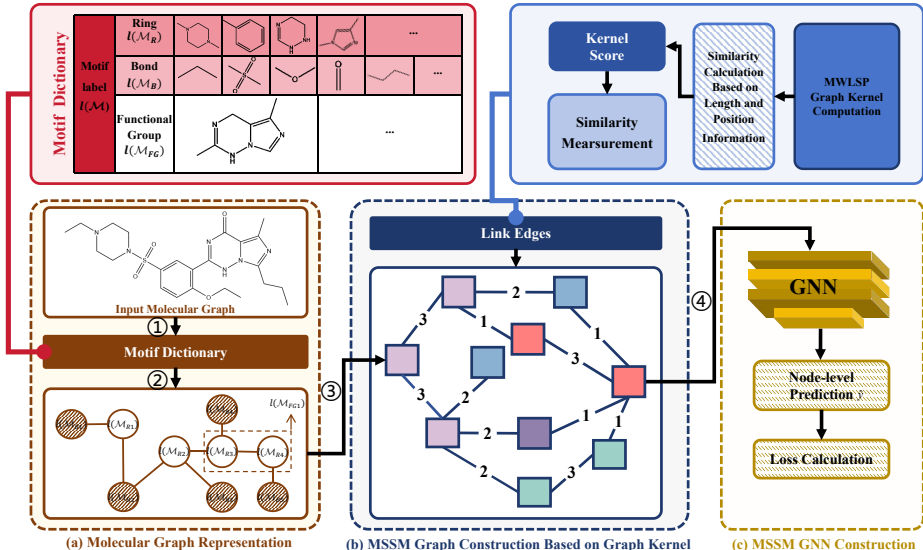


Fig. 2: The framework of our proposed Molecular Structural Similarity Motif Graph Neural Network.

Motif Dictionary Let $\mathcal{G} = (V, E)$ denote a molecular graph, where V is a set of atoms, $E \subseteq V \times V$ is a set of bonds between atoms. Generally, we denote a motif of the molecule \mathcal{G} by $\mathcal{M} = \langle \hat{V}, \hat{E} \rangle$, where \hat{V} is a subset of V and \hat{E} is the subset of E corresponding to \hat{V} , which includes all edges connecting nodes in \hat{V} . Due to the impact of ring, bond, and functional group structures on a molecule’s stability, mechanical properties, and reactivity [42], we extract these structures as three distinct types of motifs from \mathcal{G} . This extraction aims to establish a correlation between molecular structure and properties, facilitating a targeted capture of diverse chemical features within the molecule. It considers important structural components within the molecule as much as possible and can be extended to different types of molecules, making it a general approach.

To systematically organize and store the extracted motif information, we construct a motif dictionary \mathcal{D} by preprocessing all molecules in the dataset, as outlined in step ① of Fig. 2(a). The \mathcal{D} contains molecular identifiers as outer keys, each associated with nested dictionaries. These inner dictionaries categorize structural motif types with corresponding lists of extracted labels. We define the label $l(\mathcal{M})$ as the type of \mathcal{M} . The example in Fig. 2(a) illustrates that ring type Piperazine can be expressed as $l(\mathcal{M}_{R1})$. This organization efficiently stores and retrieves structural information within each molecule.

Molecular Graph Re-representation Based on the motif dictionary, we traverse the structure type and their corresponding motif lists for each molecule

a six-membered ring compound containing two nitrogen atoms

within it. This process aims to re-represent the molecular graph by establishing connections between the motifs in molecules. We defined the graph as $\mathcal{G}_{\mathcal{M}} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} denote a set of motifs and a set of adjacency relationship between motifs of the molecule, respectively. In the $\mathcal{G}_{\mathcal{M}}$, a motif \mathcal{M} is associated with a label $l(\mathcal{M})$ and adjacent motifs are connected by edges. As illustrated in step ②, for the drug molecule vardenafil, we can use the proposed algorithm to construct a $\mathcal{G}_{\mathcal{M}}$ from motifs out of the \mathcal{D} .

3.2 MSSM Graph Construction Based on Graph Kernel

Through the above method, we obtain a re-representation molecular graph $\mathcal{G}_{\mathcal{M}}$. To provide GNN with more information, we will construct a Molecular Structural Similarity Motif (MSSM) graph in step ③. In the MSSM graph, each node represents a $\mathcal{G}_{\mathcal{M}}$, and the edge represents two nodes $\mathcal{G}_{\mathcal{M}_1}$ and $\mathcal{G}_{\mathcal{M}_2}$ with structural similarity. We calculate the similarity between two $\mathcal{G}_{\mathcal{M}}$ by utilizing Mahalanobis **W**eisfeiler-**L**ehman **S**hortest-**P**ath(MWLSP) graph kernel.

The fundamental idea of the graph kernel is to measure the similarity via the comparison of $\mathcal{G}_{\mathcal{M}}$ ' substructures. The kernel we proposed retains expressivity and is still computable in polynomial time.

As depicted in Fig. 2, MWLSP graph kernel takes $\mathcal{G}_{\mathcal{M}_1}$, $\mathcal{G}_{\mathcal{M}_2}$ as input, and its main process consists of the following steps: (i) Preprocess Input; (ii) Perform MWLSP Graph Kernel Computation; (iii) Comparison Scores of Graph-substructures. We give a pseudocode description of the MWLSP Graph Kernel in Algorithm 1.

Preprocess Input In line 2, we utilize Floyd-transformation (For detailed explanation, see **Appendix A.1**) $Ft(\mathcal{G}_{\mathcal{M}})$ to convert graphs $\mathcal{G}_{\mathcal{M}_1}$ and $\mathcal{G}_{\mathcal{M}_2}$ into graphs \mathcal{G}_{F1} and \mathcal{G}_{F2} , respectively. $Ft(\mathcal{G}_{\mathcal{M}})$ generates the shortest path between all nodes in $\mathcal{G}_{\mathcal{M}}$. The shortest path between the vertex v and u is represented as (v, u) . The (v, u) is the shortest path among all paths between two nodes. \mathcal{G}_{F1} and \mathcal{G}_{F2} contain all the information regarding the shortest path substructure partitions in $\mathcal{G}_{\mathcal{M}_1}$ and $\mathcal{G}_{\mathcal{M}_2}$, respectively. Specifically, \mathcal{G}_{F1} has the same vertices as $\mathcal{G}_{\mathcal{M}_1}$, and the edge (v, u) in \mathcal{G}_{F1} represents detailed information about the shortest path in $\mathcal{G}_{\mathcal{M}_1}$.

$$\mathcal{G}_{F1} = Ft(\mathcal{G}_{\mathcal{M}_1}) \quad \mathcal{G}_{F2} = Ft(\mathcal{G}_{\mathcal{M}_2}) \quad (1)$$

Perform MWLSP Graph Kernel Computation $K_{mwlspace}(\mathcal{G}_{F1}, \mathcal{G}_{F2})$ will compute the similarity between two graphs, $\mathcal{G}_{\mathcal{M}_1}$ and $\mathcal{G}_{\mathcal{M}_2}$, by summing up $k(e_1, e_2)$ i.e., the comparison scores between substructures e_1 and e_2 in line 3-7. E'_1 is the set of all edges in \mathcal{G}_{F1} and e_1 is one of the edges in E'_1 . e_1 represents a shortest path substructure in $\mathcal{G}_{\mathcal{M}_1}$, and similarly for e_2 .

$$K_{mwlspace}(\mathcal{G}_{F1}, \mathcal{G}_{F2}) = \sum_{e_1 \in E'_1} \sum_{e_2 \in E'_2} k(e_1, e_2) \quad (2)$$

Algorithm 1: MWLSP Graph Kernel Calculation

Data: *Graphs* $\mathcal{G}_{\mathcal{M}_1} = (\mathcal{V}_1, \mathcal{E}_1)$, $\mathcal{G}_{\mathcal{M}_2} = (\mathcal{V}_2, \mathcal{E}_2)$, c , H
Result: *Kernel Score* $K_{mwls\text{p}}$

```

1 Function MWLSPGraphKernel( $\mathcal{G}_{\mathcal{M}_1}$ ,  $\mathcal{G}_{\mathcal{M}_2}$ ,  $c$ ,  $H$ ):
2    $\mathcal{G}_{F1} \leftarrow Ft(\mathcal{G}_{\mathcal{M}_1})$   $\mathcal{G}_{F2} \leftarrow Ft(\mathcal{G}_{\mathcal{M}_2})$ ;
3    $kernel\_score \leftarrow 0$ ;
4   for  $e_1$  in  $E(\mathcal{G}_{F1})$  do
5     for  $e_2$  in  $E(\mathcal{G}_{F2})$  do
6        $kernel\_score \leftarrow kernel\_score + k(e_1, e_2, c, H)$ ;
7   return  $kernel\_score$ ;

8 Function LengthSim( $e_1$ ,  $e_2$ ,  $c$ ):
9    $sim_1 \leftarrow \max(0, c - |length(e_1) - length(e_2)|)$ ;
10  return  $sim_1$ ;

11 Function PositionSim( $e_1$ ,  $e_2$ ,  $H$ ):
12  Initialize labels  $L_1$  and  $L_2$  based on  $e_1$  and  $e_2$ ;
13  for  $h$  in  $[0, H]$  do
14    for  $u$  in  $V(e_1)$  do
15       $nbrs\_sorted \leftarrow \text{sort}(\text{labels of neighbors of } u \text{ lexicographically})$ 
16       $L^{(h+1)}(u) \leftarrow \text{hash}(L^h(u), nbrs\_sorted)$ 
17    for  $v$  in  $V(e_2)$  do
18       $\text{Calculate } L^{(h+1)}(v) \text{ using the same method as above.}$ 
19    Calculate the Mahalanobis distance  $D^{(h)}(u, v)$  between
20     $L^{(h)}(u)$  and  $L^{(h)}(v)$  at the  $h$ -th iteration.
21    Sum  $D(u, v)$  across all final iteration yields  $sim_2$ 
22  return  $sim_2$ ;

23 Function  $k(e_1, e_2, c, H)$ :
24    $sim_1 \leftarrow \text{LengthSim}(e_1, e_2, c)$   $sim_2 \leftarrow \text{PositionSim}(e_1, e_2, H)$ ;
25  return  $sim_1 \times sim_2$ ;

```

Proposition 1. *Let n be the average number of nodes and d be the dimensionality of the features. Each node is associated with a d -dimensional feature vector. The time complexity for the kernel given by Eq. 2 is $O(n^3 + n^4 * (1 + Hnd^3))$.*

The proof is given in the **Appendix B**.

Comparison Scores of Graph-substructures For $k(e_1, e_2)$, we will calculate the similarity of substructures from two aspects: length and position. The calculation formulas are respectively $sim_1(e_1, e_2)$ and $sim_2(e_1, e_2)$. For the aspect of length, $sim_1(e_1, e_2)$ utilizes the Brownian bridge [6] to assess the similarity between e_1 and e_2 in line 8-10. It returns the largest value when two edges have identical lengths and 0 when the edges differ in length by more than a hyperparameter c . Furthermore, we can change the c to control the similarity threshold, thus adjusting the filtering criteria.

$$sim_1(e_1, e_2) = \max(0, c - |length(e_1) - length(e_2)|) \quad (3)$$

For the aspect of positional information, $sim_2(e_1, e_2)$ establishes a Weisfeiler-Lehman(WL) propagation scheme [31] on the graphs, iteratively comparing labels on the nodes and their neighbors via Mahalanobis Distance(MD) [8].

Specifically, we let h be the current WL iteration which ranges from 0 to H (H is the total number of iterations). $L^h(u)$ is a set of node labels, representing the positional information of node u at the current iteration h . $\mathcal{N}^h(u) = \{L^h(u_{left}), L^h(u_{right})\}$ represents the positional information of u 's neighboring nodes at the current iteration h . In the shortest path graph, u_{left} and u_{right} are the only two neighbor nodes of u . The scheme primarily consists of several steps, described in line 11-22:

Firstly, we compare two paths, e_1 and e_2 , by utilizing the motif labels to initialize the sets of all node labels on these paths in line 12.

$$L^0(u) = l(u) \quad (4)$$

Next, if identical node labels exist, further iterative evaluation is conducted. We define the iterative rule with the hash function: in each iteration, the positional information of u includes one more iteration of node connectivity compared to the previous iteration. By inputting the positional information of the current iteration's node u , i.e., $L^h(u)$ and its neighboring nodes, i.e., $\mathcal{N}^h(u)$, we use Eq. 5 to compute $L^{h+1}(u)$, i.e., the positional information of u in the next iteration $h + 1$. And $sort(\cdot)$ sorts the labels lexicographically. The specific execution process is shown in line 13-18.

$$L^{h+1}(u) = hash(L^h(u), sort((L^h(v_1), \dots, L^h(v_{|\mathcal{N}^h(u)|}))), \quad (5)$$

$$v_j \in \mathcal{N}^h(u), j \in \{1, \dots, |\mathcal{N}^h(u)|\}.$$

Through the aforementioned process, we can represent the positional information of all nodes in e_1 and e_2 by utilizing $L^h(u)$. Furthermore, in line 20, we use MD (Please see **Appendix A.2** for explanation) to measure the similarity between nodes. $D^h(u, v)$ denotes the MD between the $L^h(u)$ and $L^h(v)$ at a specific iteration h . M^h is the covariance matrix $Cov(L^h(u), L^h(v))$. The utilization of MD considers the diverse distribution characteristics of nodes belonging to different types in the heterogeneous feature space. In the context of \mathcal{G}_M , distinct types of motifs may correspond to varied structures or properties. Therefore, we can quantify the similarity between motifs based on the distribution characteristics of each motif type.

$$D^h(u, v) = \sqrt{(L^h(u) - L^h(v))^T M^h (L^h(u) - L^h(v))} \quad (6)$$

Finally, we cumulatively aggregate the MD from the 0-th to the H -th iteration in line 21. Through a weighted synthesis, we calculate the relational similarity between u and v , considering positional information across all iterations. Therefore, we can calculate the similarity score $sim_2(e_1, e_2)$ by comparing the position similarity relationships among all nodes in e_1 and e_2 .

$$sim_2(e_1, e_2) = \sum_{u \in V(e_1)} \sum_{v \in V(e_2)} \exp \left(-\frac{1}{2} \sum_{h=0}^H D^h(u, v) \right) \quad (7)$$

For the above iterative process, we set two termination conditions:

(i) There is no intersection in the positional information of all nodes in e_1 and e_2 within the current iteration. This condition implies that, in the next iteration, the positional information of nodes for e_1 and e_2 is dissimilar, so we can terminate early.

(ii) We have calculated the positional information for all iterations in e_1 and e_2 , and the total number of iterations will not exceed $\min(|\text{length}(e_1)|, |\text{length}(e_2)|)$.

In summary, the comparison score of the graph substructure can be obtained by multiplying the similarities of the above two parts in line 23-25.

$$k(e_1, e_2) = sim_1(e_1, e_2) * sim_2(e_1, e_2) \quad (8)$$

MSSM Graph Construction We can construct the MSSM graph based on the similarity calculation result of the above MWLSP graph kernel. Since the structural similarity analysis of molecules often does not require very precise numerical values, it focuses on the relative similarity between molecules. To reduce the complexity of the comparison, we simplify the kernel score to an integer range of $[0, 3]$ by dividing it by the maximum achievable value:

$$S(Ft(\mathcal{G}_{M_i}), Ft(\mathcal{G}_{M_j})) = \left\lfloor \frac{3 \cdot K_{mwls p}(Ft(\mathcal{G}_{M_i}), Ft(\mathcal{G}_{M_j}))}{\max(K_{mwls p}(Ft(\mathcal{G}_{M_i}), Ft(\mathcal{G}_{M_j})))} \right\rfloor \quad (9)$$

where $\lfloor x \rfloor$ represents rounding x down to the nearest integer.

Considering the above possible calculation results, we use the similarity score $S(Ft(\mathcal{G}_{M_i}), Ft(\mathcal{G}_{M_j}))$ to represent the corresponding edge weight value A_{ij} and formally establish detailed measurement standard Sim_{ij} as follows:

$$Sim_{ij} = \begin{cases} \textit{Very High Similarity} & \textit{if } A_{ij} = 3, \\ \textit{Relatively High Similarity} & \textit{if } A_{ij} = 2, \\ \textit{Average Similarity} & \textit{if } A_{ij} = 1, \\ \textit{Dissimilar} & \textit{if } A_{ij} = 0 \end{cases} \quad (10)$$

where if $A_{ij} > 0$, \mathcal{G}_{M_i} and \mathcal{G}_{M_j} have a similar relationship, and a connecting edge with corresponding weight value needs to be established; otherwise, there is no need to perform connection processing. Fig. 2(b) provides an example of MSSM graph construction.

3.3 MSSM-GNN Construction

In this part, we build an MSSM-GNN to learn graph structural feature representations of the MSSM graph. In graph learning, the input MSSM graphs can be denoted as $\mathcal{G}_{MSSM} = (\mathcal{V}_{MSSM}, \mathcal{E}_{MSSM})$, where \mathcal{V}_{MSSM} is the node set of $\mathcal{G}_{\mathcal{M}}$, and $\mathcal{E}_{\mathcal{MSSM}}$ is the edge set of similarity relationship between two $\mathcal{G}_{\mathcal{M}}$. And

we use $y \in \mathcal{Y}$ as the node-level property label for $\mathcal{G}_{\mathcal{M}_i}$, where \mathcal{Y} represents the label space.

For graph property prediction, a predictor with the encoder-decoder architecture is trained to encode \mathcal{G}_{MSSM} into a node representation vector in the latent space and decode the representation to predict \hat{y} . Specifically, we fed the MSSM graph data into GNN to acquire \hat{y} (corresponds to step ④):

$$\hat{y} = \text{GNN}(\mathcal{G}_{MSSM}) \in \mathcal{Y}. \quad (11)$$

The loss function used in our model is the label prediction loss. The label prediction loss function \mathcal{L}_{pred} is derived similarly to existing methods:

$$\mathcal{L}_{pred} = \text{CE}(\hat{y}, y). \quad (12)$$

where \hat{y} represents the predicted value, y is the ground truth, and CE represents the Cross-Entropy loss function used in classification tasks.

In this way, we can get a more comprehensive feature representation of the entire \mathcal{G}_{MSSM} . It contains all the information on the connected motifs, retaining the atomic structure relationships and connections within the original motifs. Therefore, we can get a more accurate prediction of molecular properties based on the MSSM-GNN. The process is illustrated in Fig. 2(c).

4 Experiments

In this section, we investigate how our proposed method improves GNN performance on molecular property tasks. In our investigations, we raise the following questions: **Q1**: Compared with state-of-the-art baselines, how effective is MSSM-GNN in improving the accuracy of molecular prediction on common bioinformatics graph benchmark datasets? **Q2**: If experiments are conducted on real-world datasets, will MSSM-GNN still have an effect? **Q3**: Does feature learning of similarities between molecules play a more critical role in MSSM-GNN? **Q4**: What impact will the setting of the similarity threshold on different datasets have on the final classification results?

In response to the above problems, we conducted a series of experimental studies. Some basic settings of experiments and analysis of results are as follows:

4.1 Experimental Settings

Datasets. To verify whether MSSM-GNN provides more information conducive to accurate classification, we evaluate our model on five popular bioinformatics graph benchmark datasets from TUDataset [27], which includes four molecular datasets PTC [34], MUTAG [9], NCI1 [35], MUTAGENICITY [20] and one protein dataset PROTEINS [3].

Table 1: Graph classification accuracy (%) on various TUDataset graph classification tasks. The best performers on each dataset are shown in **bold**.

Methods	PTC	NCI1	MUTAG	PROTEINS	MUTAGENICITY
DGCNN	58.6±2.5	74.4±0.5	85.8±1.7	75.5±0.9	72.3±2.6
GCN	64.2±4.3	80.2±2.0	85.6±5.8	76.0±3.2	79.8±1.6
GIN	64.6±7.0	82.7±1.7	89.4±5.6	76.2±2.8	82.0±0.3
PatchySAN	60.0±4.8	78.6±1.9	92.6±4.2	75.9±2.8	77.9±1.3
GraphSAGE	63.9±7.7	77.7±1.5	85.1±7.6	75.9±3.2	78.8±1.2
PPGN	66.2±6.5	83.2±1.1	90.6±8.7	77.2±4.7	78.6±0.9
WEGL	64.6±7.4	76.8±1.7	88.3±5.1	76.1±3.3	80.8±0.4
CapsGNN	71.2±1.9	78.4±1.6	86.7±6.9	76.3±4.6	79.5±0.7
GSN	68.2±7.2	83.5±2.3	90.6±7.5	76.6±5.0	81.0±1.5
HM-GNN	78.5±2.6	83.6±1.5	96.3±2.8	79.9±3.1	83.0±1.1
GPNN	78.2±1.2	83.1±0.3	92.6±1.8	76.8±3.9	83.0±0.4
OURS	81.1±1.7	85.5±0.3	97.3±2.6	83.3±0.4	84.0±0.5

Baselines. We compare our model with eleven state-of-the-art GNN models for molecular property tasks: Deep Graph CNN (DGCNN) [29], GCN [21], GIN [40], PATCHYSAN [28], GraphSAGE [14], Provably Powerful Graph Networks (PPGN) [24], Wasserstein Embedding for Graph Learning (WEGL) [22], Capsule Graph Neural Network (CapsGNN) [39], GSN [4], HM-GNN [45], GPNN [15].

4.2 Performance Evaluation on Molecular Graph Datasets

To learn graph feature representations in our molecular structural similarity motif graphs, 3 GNN layers are applied. For a fair comparison, we evaluate all baselines using the experiment settings provided by [45]. The hyper-parameters we tune for each dataset are (1) the learning rate \in 0.01, 0.05; (2) the number of hidden units \in 16, 64, 1024; (3) the dropout ratio \in 0.2, 0.5. We set the verification method as the mean and standard deviation of the seven best validation accuracies from ten folds. We compare MSSM-GNN with the baseline approaches on the abovementioned dataset to answer **Q1**. The comparison results are summarized in Table 1. We make the following observations:

MSSM-GNN significantly outperforms baseline models on all five datasets for molecular prediction. Among them, on the PROTEINS dataset, the accuracy of MSSM-GNN increased by 3.4% compared with the best method. The superior performances on five molecular datasets demonstrate that motif substructures extracted from the motif dictionary, along with the calculated similarity relationships between molecular nodes based on it, facilitate GNNs in learning improved motif-level and molecular-level feature representations of molecular graphs.

Table 2: Graph Classification Results (%) on Open Graph Benchmark datasets.

Methods	ogbg-molhiv	ogbn-proteins	ogbg-moltoxcast	ogbg-molpcba
GCN	75.99 \pm 1.19	72.51 \pm 0.35	61.13 \pm 0.47	24.24 \pm 0.34
GIN	77.07 \pm 1.49	77.68 \pm 0.20	62.19 \pm 0.36	27.03 \pm 0.23
GSN	77.90 \pm 0.10	85.80 \pm 0.28	62.61 \pm 0.45	27.00 \pm 0.70
PNA	79.05 \pm 1.32	86.82 \pm 0.18	63.47 \pm 0.67	25.70 \pm 0.60
HM-GNN	79.03 \pm 0.92	86.42 \pm 0.08	64.38 \pm 0.39	28.70 \pm 0.26
GPNN	77.70 \pm 2.30	87.74 \pm 0.13	65.22 \pm 0.47	28.90 \pm 0.91
OURS	79.70\pm0.03	89.17\pm0.07	66.57\pm1.00	30.07\pm0.37

4.3 Performance Evaluation on Large-Scale Real-World Datasets

To answer **Q2**, we evaluate our model on four large-scale real-world datasets from the Open Graph Benchmark (OGB) [16]. They are two binary classification datasets—ogbg-molhiv, ogbn-proteins and two multiclass classification datasets—ogbg-moltoxcast, ogbg-molpcba.

In this part, we compare our model with GIN, GCN, GSN, PNA, HM-GNN and GPNN. Except that the hyperparameters we tuned for each dataset varied as (1) learning rate \in 0.01, 0.001; (2) number of hidden units \in 10, 16; (3) dropout rate \in 0.5, 0.7, 0.9; (4) the batch size \in 128, 5000, 28000, others are the same as above experiment. Table 2 shows the AP results on Ogbg-molpcba and ROC-AUC results on the other three datasets. We observe: *our method is significantly better than the other compared methods by obvious margins*. The results prove our model’s superior generalization ability on real-world datasets, which is crucial for its potential applications in various domains, including drug discovery, bioinformatics, and chemical safety assessment.

4.4 Ablation Study

To address **Q3**, we conducted ablation experiments on different components of MSSM-GNN, focusing on the motif-based molecular graph representation and the similarity calculation. The corresponding conclusions are as follows:

Effect of Motif-Dictionary Representation As shown in Table 3, comparing task performance before and after removing the motif dictionary module yields the following observations: Performance on three graph classification datasets benefits from the module, resulting in accuracy improvements ranging from 0.8% to 2.8%. These improvements could potentially be attributed to the module’s effective learning of valuable information about the molecule’s substructure.

Effect of Length-Similarity Calculation As shown in Table 3, we observe a significant drop in performance when the length-similarity calculation is not

Table 3: Ablation studies of the motif dictionary and measurement of length and position similarity.

Datasets	PTC_MR	PTC_FR	MUTAG	PROTEINS
MSSM-GNN	81.1±1.7	80.9±1.5	97.3±2.6	83.3±0.4
w/o motif dictionary	78.3±1.1	78.6±1.4	96.5±2.7	82.5±1.2
w/o length similarity	77.9±1.5	78.3±1.1	94.6±0.2	81.2±0.9
w/ edit distance	77.1±2.9	78.2±1.7	93.1±0.6	80.8±0.4

included, amounting to an absolute drop of 2.1% - 3.2%. These observations confirm that evaluating path structures from a length perspective indeed facilitates the learning of the global information and inherent connectivity relationships among motif-level substructures, thereby contributing to representing graph information more comprehensively.

Effect of Position-Similarity Calculation In MSSM-GNN, the location similarity calculation method we designed is MWL. By replacing MWL with edit distance, we examined the impact of the graph similarity metric. As Table 3 shows, MWL offers advantages over edit distance. MWL not only quantifies structural similarity but also incorporates the type and position information of different nodes in graph modeling, thereby effectively representing the real molecular graph structure. Meanwhile, MWL becomes particularly advantageous for larger-scale graph datasets, offering significant enhancements by extracting richer structural information. For example, our model enhances PROTEINS more than MUTAG.

4.5 Sensitive Analysis

In this part, to explore Q4, we further evaluate the hyperparameter c that we introduced in our proposed similarity calculation formula. We modify the value of hyperparameter c that controls the similarity threshold and observe how the performance changes. We perform such experiments on multiple datasets. The results are shown in Fig. 3.

As observed, the performance peaks when the value of c is 2 across all three datasets. With the increase in c , the impact of the similarity threshold on training also becomes more pronounced. It is evident that the performance of MSSM-GNN decreases as c increases from 2 to 6, indicating that the c indeed influences the representation learning capabilities of MSSM-GNN. We believe that c assists in filtering out samples with low similarity, emphasizing those contributing more significantly to the training, thereby enhancing overall performance.

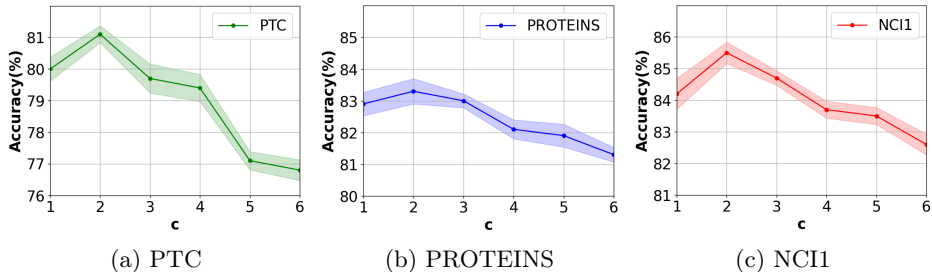


Fig. 3: Performance of MSSM-GNN on three different datasets with varying hyperparameters c .

5 Conclusions

This paper proposes an effective model for molecular graph representation learning, Molecular Structural Similarity Motif GNN (MSSM-GNN). We explicitly incorporate the similarity representations between molecules into GNN and jointly update them with motif representations. Specifically, we connect two molecules through edge weights calculated by a novel MWLSP graph kernel, enabling message passing between molecular graphs. We use the GNN model to learn the MSSM graph and get the motif-level and molecule-level graph embedding. Experiments demonstrate the superiority of our model in various datasets, which beats a group of baseline algorithms.

Acknowledgments. The authors would like to thank the editors and reviewers for their valuable comments. This work is supported by the CAS Project for Young Scientists in Basic Research (Grant No. YSBR-040) and the National Natural Science Foundation of China (Grant No. 62372439), the Natural Science Foundation of Beijing, China (Grant No. 4232038), the Basic Research Program of ISCAS (Grant No. ISCAS-JCMS-202307).

References

1. Atsango, A., Diamant, N.L., Lu, Z., Biancalani, T., Scalia, G., Chuang, K.V.: A 3d-shape similarity-based contrastive approach to molecular representation learning. arXiv preprint [arXiv:2211.02130](https://arxiv.org/abs/2211.02130) (2022)
2. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: Fifth IEEE international conference on data mining (ICDM’05). pp. 8–pp. IEEE (2005)
3. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S., Smola, A.J., Kriegel, H.P.: Protein function prediction via graph kernels. *Bioinformatics* **21**(suppl_1), i47–i56 (2005)
4. Bouritsas, G., Frasca, F., Zafeiriou, S., Bronstein, M.M.: Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(1), 657–668 (2022)

5. Cantoni, V., Gatti, R., Lombardi, L.: Morphological analysis of 3d proteins structure. In: International Conference on Bioinformatics Models, Methods and Algorithms. vol. 2, pp. 15–21. SciTePress (2011)
6. Chow, W.C.: Brownian bridge. Wiley interdisciplinary reviews: computational statistics **1**(3), 325–332 (2009)
7. Dan, J., Wu, R., Liu, Y., Wang, B., Meng, C., Liu, T., Zhang, T., Wang, N., Fu, X., Li, Q., et al.: Self-supervision meets kernel graph neural models: From architecture to augmentations. In: 2023 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 1076–1083. IEEE (2023)
8. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L.: The mahalanobis distance. Chemometrics and intelligent laboratory systems **50**(1), 1–18 (2000)
9. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry **34**(2), 786–797 (1991)
10. Degen, J., Wegscheid-Gerlach, C., Zaliani, A., Rarey, M.: On the art of compiling and using ‘drug-like’ chemical fragment spaces. ChemMedChem: Chemistry Enabling Drug Discovery **3**(10), 1503–1507 (2008)
11. Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. Advances in neural information processing systems **28** (2015)
12. Gao, H., Ji, S.: Graph u-nets. In: international conference on machine learning. pp. 2083–2092. PMLR (2019)
13. Geng, Z., Xie, S., Xia, Y., Wu, L., Qin, T., Wang, J., Zhang, Y., Wu, F., Liu, T.Y.: De novo molecular generation via connection-aware motif mining. arXiv preprint [arXiv:2302.01129](https://arxiv.org/abs/2302.01129) (2023)
14. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
15. Hevaphithige, A., Wang, Q.: Uplifting the expressive power of graph neural networks through graph partitioning. arXiv preprint [arXiv:2312.08671](https://arxiv.org/abs/2312.08671) (2023)
16. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems **33**, 22118–22133 (2020)
17. Huang, K., Xiao, C., Hoang, T., Glass, L., Sun, J.: Caster: Predicting drug interactions with chemical substructure representation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 702–709 (2020)
18. Inae, E., Liu, G., Jiang, M.: Motif-aware attribute masking for molecular graph pre-training. arXiv preprint [arXiv:2309.04589](https://arxiv.org/abs/2309.04589) (2023)
19. Jin, W., Barzilay, R., Jaakkola, T.: Hierarchical generation of molecular graphs using structural motifs. In: International conference on machine learning. pp. 4839–4848. PMLR (2020)
20. Kazius, J., McGuire, R., Bursi, R.: Derivation and validation of toxicophores for mutagenicity prediction. Journal of medicinal chemistry **48**(1), 312–320 (2005)
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2016)
22. Kolouri, S., Naderializadeh, N., Rohde, G.K., Hoffmann, H.: Wasserstein embedding for graph learning. arXiv preprint [arXiv:2006.09430](https://arxiv.org/abs/2006.09430) (2020)
23. Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., Ji, S.: Spherical message passing for 3d molecular graphs. In: International Conference on Learning Representations (ICLR) (2022)