APSIPA Transactions on Signal and Information Processing, 2025, 14, e2

Original Paper

Molecular Representation Learning via Hierarchical Graph Transformer

Zehua Wang[1,2], Yang Liu[1] and Wei Hu[1*]

[1] *Wangxuan Institute of Computer Technology, Peking University, Beijing, China*
[2] *Academy for Advanced Interdisciplinary Studies, Peking University, Beijing, China*

ABSTRACT

Molecular Representation Learning (MRL) is widely applied in various downstream tasks, such as molecule generation, molecular property prediction and reaction prediction. Nevertheless, MRL faces several challenges posed by the vast chemical space and limited labeled-data availability. In this paper, we propose Hierarchical Graph Transformer (HieGT), integrating atom-level and motif-level representations to capture local-global characteristics of molecules over a hierarchical graph. Leveraging Atom-wise Graph Attention and Motif-wise Graph Attention, HieGT enhances intrinsic representation understanding of molecules. The proposed method achieves state-of-the-art performance over the molecular property prediction benchmark *PCBA*, and competitive results on *PCQM4Mv2* with better interpretability.

*Corresponding author: Wei Hu (email: forhuwei@pku.edu.cn)

Received 04 November 2024; revised 23 December 2024; accepted 25 January 2025
ISSN 2048-7703; DOI 10.1561/116.20240082

1 Introduction

Molecular Representation Learning (MRL) aims to utilize machine learning to encode molecules as numerical feature vectors for downstream applications, such as molecular property prediction [39], reaction prediction [31], drug design [32] and drug-drug interaction prediction [42]. However, the chemical space of molecules are extremely vast with limited labeled data, which poses great challenge to extract effective representations.

There are three main manners to represent molecules: the fingerprint, sequence and graph, as shown in Fig. 1 (a)-(c). The fingerprint [29] embodies hand-crafted information from molecules with a fixed length, which is not flexible for various tasks. The sequence [35] mainly refers to Simplified Molecular Input Line-Entry System (SMILES), which consists of ASCII strings to describe molecules. Though SMILES can be processed by Natural Language Processing methods, they are typically difficult to understand intuitively and may encounter ambiguity [15]. In contrast, the graph is a natural way to represent the topology of molecules. Typically the atoms are treated as nodes while the bonds are treated as edges, then additional information can be incorporated in nodes and edges from atom and bond features. Thus, graphs are widely adopted in recent studies [15, 44, 5, 21, 40, 24, 25, 23, 13]. In applications such as drug-drug interaction [43], drug-target binding affinity prediction [12] and biochemical reactions [36], edge directionality may be meaningful for learning asymmetric relationships between molecules. However, as we mainly focus on independent and static molecular representations rather than molecular interactions and dynamics, the chemical bonds between atoms within molecular graphs can be regarded as symmetric edges, on account of which we represent molecules as undirected graphs as in previous works [40, 24, 23, 13].

Previous works in graph-based molecular representation learning can be divided into three classes [7]: molecular-topology-based methods [15, 44], knowledge-graph-based methods [5], and spatial-learning-based methods [21, 4, 8]. Molecular-topology-based methods focus on the topological structures or substructures. However, most quantum chemical properties are derived from 3D conformations [23], which cannot be reflected from 2D topologies. Knowledge-graph-based methods extract molecular representations by knowledge graphs rather than

molecular graphs. Nevertheless, knowledge graphs relies heavily on hand-crafted features and domain knowledge, and may discard large amounts of structural features. Spatial-learning-based methods focus more on 3D geometric features of molecules. Since 3D geometric features serve as a vital role in predicting molecular properties, we choose to integrate molecular-topology-based and spatial-learning-based paradigms to extract more comprehensive molecular representation combining both 2D & 3D graph features.

As one of the most powerful models to learn molecular topological and spatial representations, graph Transformer [41] has recently gained state-of-the-art performance on many MRL tasks [40, 24, 25, 23, 13], due to its distinct self-attention mechanism for capturing long-range structural dependencies. Previous works in graph Transformer [40, 24, 23] mainly focus on the global information flow among node representations. Edge representations are merely utilized as a bias term to the attention module. To emphasize the significance of edge representations equal to node representations, the Edge-augmented Graph Transformer (EGT) [14] introduced dynamic edge channels that are updated across layers, enabling information flow between node and pair representations. Furthermore, TGT-At [13] enabled direct communication between two adjacent pairs in a graph via novel triplet attention and aggregation mechanisms.

Though the atom-level global graph structure has been effectively exploited in previous works, the local context of molecules is not fully studied yet in the networks. It has been found that molecules can be characterized by a set of *motifs*, each of which may correspond to a certain type of local substructures and functions (similar to chemical functional groups) [34]. As illustrated in Fig. 2, the sample molecule can be composed of four motifs, such as the benzene ring and the carbonyl group (the 3rd and 4th colored circle from left to right). Accordingly, the molecule can be represented as a motif-wise graph, embodying a hierarchical structure. Hence, we propose Hierarchical Graph Transformer (HieGT), the first hierarchical Transformer framework to learn molecular representation, to the best of our knowledge. We incorporate both atom-level and motif-level graph representations and come up with a hierarchical graph encoding strategy to shed new light on local-global characteristics of molecules.

In particular, we firstly decompose the atom-wise molecular graph

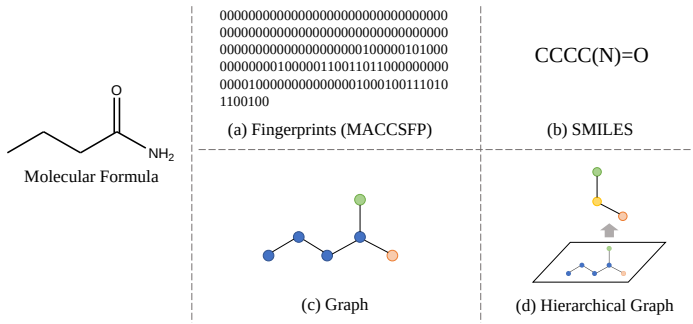

Figure 1: Representations of molecules. (a) Fingerprints calculated as MACCSFP by rdkit [18] in binary. (b) The SMILES. (c) The graph. (d) The proposed hierarchical graph representation, where molecules can be characterized by a set of motifs. Each motif may correspond to a certain type of local substructures and functions.

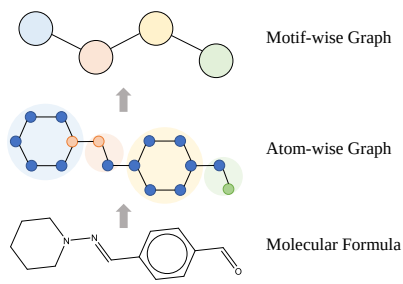

Figure 2: The atom-wise graph could be decomposed into the motif-wise graph by certain rules.

into motifs based on three hand-crafted rules, as specified in Sec. 4.1. Then the motifs are assembled with the same connectedness, composing the motif-wise graph, as illustrated in Fig. 2. The atom-wise and the motif-wise graphs are both exploited in the proposed method in a hierarchical manner, as illustrated in Fig. 1(d).

Then, we propose to learn molecular representations based on two natural assumptions. First, there exists information exchange among motifs via inter-motif edges. Second, the information flow among atoms in the same motif are regulated by the motif via intra-motif edges. Based on these two assumptions, we develop two procedures to learn Atom-wise Graph Attention (AGA) via intra-motif edges, as illustrated in Fig. 3(a), and Motif-wise Graph Attention (MGA) via inter-motif

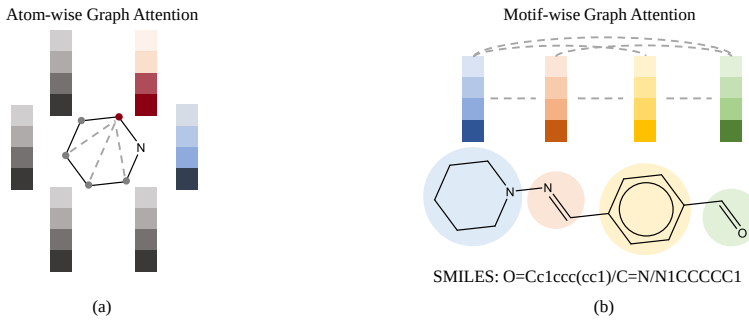

Figure 3: Atom-wise Graph Attention (AGA) and Motif-wise Graph Attention (MGA).

edges, as illustrated in Fig. 3(b). After AGA and MGA, the learned molecular representations are projected for downstream tasks.

Our main contributions are summarized as follows:

- We propose a novel hierarchical molecular representation learning paradigm (HieGT), which integrates both atom-level and motif-level information.
- We introduce Atom-wise Graph Attention and Motif-wise Graph Attention to learn the information flow within and among motifs, by constraining attention over intra-motif and inter-motif edges.
- Experimental results demonstrate that our approach establishes a new state-of-the-art (SOTA) on the PCBA dataset [11] and achieves competitive performance with the SOTA on the PCQM4Mv2 dataset [10] while obtaining better interpretability.

2 Related Works

Previous works in graph-based MRL methods can be divided into three classes [7]: molecular-topology-based methods, knowledge-graph-based methods, and spatial-learning-based methods.

Molecular-topology-based methods. Molecular-topology-based methods focus on the topological structures or substructures

of the molecular graphs. Jin *et al.* [15] generate molecular graphs from SMILES strings by the junction tree variational autoencoder for molecular graph generation. Zhang *et al.* [44] extract motifs from molecular graphs and design a self-supervised motif generation framework for molecular property prediction. However, motif features are used without explicit atom features, and 2D topology merely reveals the connectivity between atoms, while the actual 3D distances may significantly vary between different atom pairs sharing the same local topology.

Knowledge-graph-based methods. Knowledge-graph-based methods extract molecular-structure-invariant knowledge. KCL [5] adopts contrastive learning with an external knowledge graph, which is formed by triples in the form of (chemical element, relation, attribute). Nevertheless, atom pair-wise characteristics are largely neglected, especially quantitative features such as geometric features.

Spatial-learning-based methods. Spatial-learning-based methods pay more attention to 3D geometric features of molecules. GeomGCL [21] proposes graph contrastive learning by embedding distances and angles across 2D and 3D views. The properties of molecules are mostly determined by their 3D structures [4, 8], which explains why spatial-learning-based methods typically achieve better performance on MRL tasks.

By combining molecular-topology-based and spatial-learning-based paradigms, the graph Transformer model has recently achieved state-of-the-art performance across numerous downstream tasks in MRL with its unique self-attention mechanism. Graphormer [40] encodes the centrality, shortest path distance and edge features into the standard Transformer architecture. Nevertheless, only 2D topological information is encoded. Transformer-M [24] develops two separated channels to encode both 2D and 3D structural information and incorporates them with the atom features in the network modules. GPS++ [25] is a hybrid Message Passing Neural Network (MPNN) and Transformer to incorporate 3D atom positions and an auxiliary denoising task. Uni-Mol+ [23] generates an initial molecule conformation from simple methods such as RDKit [18], and iteratively updates the conformation, which will be used to further predict molecular properties. TGT-At [13] enables direct communication between two adjacent pairs in a graph via novel

triplet attention and aggregation mechanisms. Our method adopts the transformer architecture but incorporates both atom-level and motif-level features in a hierarchical manner to extract local-global intrinsic molecular representation.

3 Preliminary

3.1 Graph Neural Networks (GNN)

We represent molecules on undirected graphs. An undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is composed of a node set $\mathcal{V}$ with cardinality $|\mathcal{V}| = N$, and an edge set $\mathcal{E}$ connecting nodes. The typical GNNs iteratively update the representation of a node v_i by aggregating representations of its neighbors:

$$\mathbf{h}_i^{l+1} = \text{UPDATE}(\mathbf{h}_i^l, \text{AGGREGATE}(\{\mathbf{h}_j^l\}_{j \in \mathcal{N}(v_i)})), \tag{1}$$

where $\mathbf{h}_i^l$ is the representation of v_i at the l-th layer, and $\mathcal{N}(v_i)$ is the set of neighbors of v_i.

3.2 Transformer

The Transformer model consists of Transformer layers [33], each of which includes a self-attention module and a position-wise feed-forward network (FFN).

Multi-head self-attention. Denote $\mathbf{H}_l$ as the input of self-attention module on the transformer layer l, and d as the hidden dimension, one head k of self-attention $\mathbf{A}_k(\mathbf{H})$ is represented as:

$$\mathbf{Q}_k = \mathbf{H}_l \mathbf{W}_{Q_k}, \mathbf{K}_k = \mathbf{H}_l \mathbf{W}_{K_k}, \mathbf{V}_k = \mathbf{H}_l \mathbf{W}_{V_k}, \tag{2}$$

$$\mathbf{A}_k(\mathbf{H}_l) = \text{softmax}(\frac{\mathbf{Q}_k \mathbf{K}_k^\top}{\sqrt{d}} + \mathbf{B}_k)\mathbf{V}_k, \tag{3}$$

where $\mathbf{W}_{Q_k}$, $\mathbf{W}_{K_k}$ and $\mathbf{W}_{V_k}$ are learnable projection matrices, and $\mathbf{B}_k$ is the graph structural encodings below as the attention bias of the k heads.

Let $\mathbf{W}_k$ be the learnable projection matrix to map the concatenated output of all h heads, the multi-head self-attention is denoted as:

$$\mathbf{A}(\mathbf{H}_l) = \text{CONCAT}(\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_h)\mathbf{W}_l. \tag{4}$$

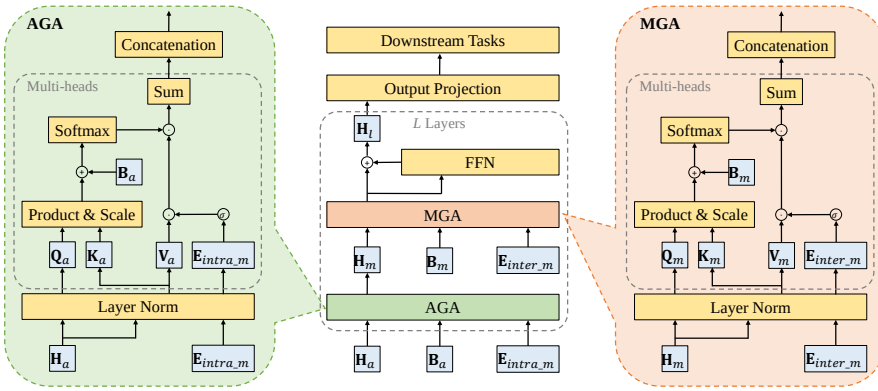

Figure 4: The framework of the proposed Hierarchical Graph Transformer.

Transformer layer. After the multi-head self-attention, the feed-forward network (FFN) is applied, which is composed of a pre-norm layer, a linear transformation layer, a non-linear activation layer, followed by another linear transformation layer. As in previous works [40, 13], we choose the widely used Gaussian Error Linear Unit (GELU) [9] as the activation function. Both the output of self-attention and FFN of layer l is processed with pre-norm layer normalization and residual connection:

$$\mathbf{H}'_l = \text{LayerNorm}(\mathbf{H}_l + \mathbf{A}(\mathbf{H}_l)), \tag{5}$$

$$\mathbf{H}_{l+1} = \text{LayerNorm}(\mathbf{H}'_l + \text{FFN}(\mathbf{H}'_l)), \tag{6}$$

where $\mathbf{H}_{l+1}$ is the output of the transformer layer $l+1$.

4 The Proposed Hierarchical Graph Transformer

We propose a Hierarchical Graph Transformer framework, as illustrated in Fig. 4. Our molecular representation learning consists of four steps: 1) hierarchical graph construction, where we design three hand-crafted rules to decompose the atom-wise graph into a motif-wise graph; 2) Atom-wise Graph Attention, which computes self-attention within motifs via intra-motif edges; 3) Motif-wise Graph Attention, which computes self-attention between motifs via inter-motif edges; and 4) output projection, which obtains representations of the whole graph by linear transformation. We elaborate on these steps in the following.

Table 1: Input atomic features and the corresponding rdkit functions.

Features	Rdkit Functions
the atomic number	GetAtomicNum()
the chiral tag	GetChiralTag()
the degree	GetTotalDegree()
the formal charge	GetFormalCharge()
the number of connected Hs	GetTotalNumHs()
the number of radical electrons	GetNumRadicalElectrons()
hybridization	GetHybridization()
aromaticity	GetIsAromatic()
whether is in ring	IsInRing()

Table 2: Input bond features and the corresponding rdkit functions.

Features	Rdkit Functions
the bond type	GetBondType()
the bond stereo	GetStereo()
whether is conjugated	GetIsConjugated()

4.1 Hierarchical Graph Construction

Graph motifs are frequently-occurring subgraph patterns (*e.g.*, functional groups of molecules), which are fundamental for both the structure and function of molecules. For instance, the benzene ring is one of the most typical motifs among molecules, which embodies special chemical properties not reflected by individual atoms. Therefore, to better extract molecular features, we propose to construct atom-wise graph and motif-wise graph to learn both local and global molecular representations.

The atom-wise graph. Following previous works [40, 24, 13], we use the input atomic and bond features as calculated by the OGB [11] Python library. Specifically, they can be represented in Table 1 and Table 2 by rdkit functions. We treat atoms as nodes and bonds as edges, then the input atomic and bond features are projected via a learnable embedding layer into node embeddings $\mathbf{X}_a$ and edge embeddings $\mathbf{E}_a$.

The motif-wise graph. To obtain motifs from a molecule in a universal manner, we consider bridge bonds that connect motifs based

on three rules:

Rule 1: The bonds that connect rings and chains (non-ring subgraphs);

Rule 2: The bonds in chains that connect carbon and non-carbon atoms;

Rule 3: The bonds in chains whose types are not single (*i.e.*double or triple).

By breaking bridge bonds, a molecule is transformed from an atom-wise graph to a motif-wise graph, where each motif is represented as a node, and the bridge bonds serve as the inter-motif edges. The new node embeddings of the motif-wise graph are obtained via Atom-wise Graph attention discussed in Sec. 4.2, while the new edge embeddings are part of the atom-wise-graph edge embeddings, merely retaining embeddings of inter-motif edges.

4.2 Atom-wise Graph Attention (AGA)

Given the atom-wise graph, we can obtain the AGA via intra-motif edges $\mathcal{E}_{intra\_m}$, illustrated as the module painted light green in Fig. 4.

As the self-attention mechanism described in Sec. 3.2 only calculates the context between each node and all the other nodes, much structural information of a graph is neglected, such as the relation between node pairs. Therefore, we introduce three graph structural encodings of the previous work [40]: centrality encoding, path encoding and edge encoding.

Centrality encoding. Node centrality measures how important a node is in the graph. Because degree centrality is one of the standard centrality measures in literature, the degree encoding of node v_i is defined as:

$$\Psi_i^{Centrality} = z^-_{deg^-(v_i)} + z^+_{deg^+(v_i)}, \tag{7}$$

where $z^-_{deg^-(v_i)}, z^+_{deg^+(v_i)} \in \mathbb{R}^d$ denote embedding vectors with indegree $deg^-(v_i)$ and outdegree $deg^+(v_i)$ respectively. For undirected graphs like molecular graphs, $deg^-(v_i)$ and $deg^+(v_i)$ are equal.

Path encoding. In the Transformer architecture, positional dependency is generally encoded as bias terms to encode global structural information. To extract positional dependency between atom pairs of a molecule, encoding the distance is the most natural way. To avoid am-

biguity with the 3D distance encoding, we apply 2D distance encoding for each connected atom pair (v_i, v_j) but rename it to path encoding:

$$\Psi_{ij}^{Path} = b_{SPD(v_i,v_j)}, \tag{8}$$

where $b_{SPD(v_i,v_j)}$ is a learnable scalar indexed by the Shortest Path Distance (SPD) between v_i and v_j.

Edge encoding. For molecular graphs, edge features represent critical properties of bonds between connected atoms. For each connected atom pair (v_i, v_j), the edge encoding is defined as:

$$\Psi_{ij}^{Edge} = \frac{1}{N} \sum_{n=1}^{N} x_{e_n} (w_n)^T, \tag{9}$$

where x_{e_n} is the feature of the n-th edge e_n in the shortest path between v_i and v_j, and w_n is the n-th weight embedding of the same dimension as x_{e_n}.

Though the above encoding methods effectively represent 2D structural features of molecular graphs, the 3D geometric features are neglected, which are more practical for discovering the actual properties of molecules. Therefore, we introduce the 3D distance encoding as [24], which is naturally invariant to translation and rotation of the 3D molecular graphs.

3D distance encoding. While path encoding can effectively represent topological distances, it fails to take the actual 3D Euclidean distances into account. Due to the complex conformation of a molecule, one atom can be close to another atom (which means a short 3D Euclidean distance) but have a long SPD. Hence, we apply 3D distance encoding as a complement for path encoding, similar to the one used in Transformer-M [24] and TGT [13]:

$$\phi_{ij}^k = \frac{1}{\sqrt{2\pi} \cdot |\sigma^k|} \exp\left[-\frac{1}{2} \left(\frac{m_{ij}^k \cdot d_{ij} + b_{ij}^k - \mu^k}{|\sigma^k|} \right)^2 \right], \tag{10}$$

where d_{ij} is the 3D Euclidean distance between atoms i and j. m_{ij}^k, b_{ij}^k are learnable scalars indexed by the pair of atom types, and μ^k, σ^k are learnable parameters for the k-th kernel ($k = 1, ..., K$ where K is the number of Gaussian Basis kernels [30]). Denoting ϕ_{ij} as the

concatenation of the outputs of all kernels, the 3D distance encoding of pair (i, j) is defined as:

$$\Psi_{ij}^{Distance} = \text{GELU}(\phi_{ij} W_{d1}) W_{d2}, \tag{11}$$

where W_{d1} and W_{d2} are learnable weight matrices, and GELU is the Gaussian Error Linear Unit [9] as the activation function.

The path encoding, edge encoding, and 3D distance encoding are combined as the atom-wise attention bias:

$$\mathbf{B}_a = \Psi^{Path} + \Psi^{Edge} + \Psi^{Distance}. \tag{12}$$

For simplicity, we omit the notation of multi-heads. Inspired by EGT [14], we use the intra-motif edge embeddings $\mathbf{E}_{intra\_m}$ to gate the information flow between atoms. Given node representations $\mathbf{H}_a$, the attention matrix $\mathbf{A}_a$ in the atom-wise graph is denoted as:

$$\mathbf{Q}_a = \mathbf{H}_a \mathbf{W}_Q, \mathbf{K}_a = \mathbf{H}_a \mathbf{W}_K, \mathbf{V}_a = \mathbf{H}_a \mathbf{W}_V, \tag{13}$$

$$\mathbf{A}_a = \text{softmax}(\frac{\mathbf{Q}_a \mathbf{K}_a^\top}{\sqrt{d}} + \mathbf{B}_a) \odot \sigma(\mathbf{E}_{intra\_m}) \mathbf{V}_a. \tag{14}$$

The network outputs the embedded representation of each atom for the MGA module.

4.3 Motif-wise Graph Attention (MGA)

Given the motif-wise graph, we can obtain the MGA via inter-motif edges $\mathcal{E}_{inter\_m}$, illustrated as the module painted light orange in Fig. 4.

In this module, we deploy a similar network framework as in the AGA module, but choose different encodings. As the motif-wise graph possesses new topological structure and edges, we retain the above manners of computing the path encoding $\Psi^{Path'}$ and edge encoding $\Psi^{Edge'}$, and combine them as the motif-wise attention bias:

$$\mathbf{B}_m = \Psi^{Path'} + \Psi^{Edge'}. \tag{15}$$

Nevertheless, the distances between motifs do not make much sense due to the nonexistence of explicit centers in each motif, so we discard

them in the module. Similar to AGA, the attention matrix $\mathbf{A}_m$ in the motif-wise graph is denoted as:

$$\mathbf{A}_m = \text{softmax}(\frac{\mathbf{Q}_m \mathbf{K}_m^\top}{\sqrt{d}} + \mathbf{B}_m) \odot \sigma(\mathbf{E}_{inter\_m})\mathbf{V}_m. \tag{16}$$

The network outputs the embedded representation of each atom for output projection.

Despite the similarity in the network architecture of AGA and MGA in Fig. 4, there are some differences in the specific implementation of AGA and MGA. First, the input data are different. AGA takes the node representations $\mathbf{H}_a$ as input, while MGA takes the output of AGA as input. Second, the atom-wise attention bias $\mathbf{B}_a$ in AGA and the motif-wise attention bias $\mathbf{B}_m$ in MGA are calculated differently, as described in Eq. 12 and Eq. 15. Third, the information flows of AGA and MGA are gated by intra-motif edges and inter-motif edges, respectively.

4.4 Output Projection

In this module, the output representation from the motif module is projected as the overall graph representation by linear transformation for downstream tasks.

5 Experiments

5.1 Experimental Setup

We follow the experimental settings in previous works [40, 24, 13]. First, we pre-train our model on the large quantum chemistry datasets *PCQM4Mv2* from OGB Large-Scale Challenge [10]. After pre-training, we implement finetuning on *PCBA* [11] for the classification task.

PCQM4Mv2. PCQM4Mv2 is a quantum chemistry dataset originally curated under the PubChemQC project [26]. The total number of training samples is 3.37 million. The task of PCQM4M-LSC is to predict HOMO-LUMO energy gap of molecules calculated by density functional theory (DFT) [3] with their 2D molecular graphs, which is one of the most practically-relevant quantum chemical properties of

molecule science [10]. We utilize the same dataset-division manner as [10] for fair comparison.

PCBA. *PCBA* is a molecular property prediction dataset with 437,929 molecules. We follow MoleculeNet [37] to split datasets into the training, validation and test set with a 80/10/10 ratio.

The experiments are conducted over eight RTX3090 (24GB RAM). We employ 24 transformer layers, and the dimension of hidden layers and feed-forward layers is set to 768. The number of attention heads is set to 32.

5.2 Pre-training

The model is pre-trained on the training set of *PCQM4Mv2*. The training set provides 3D structural information for training molecules computed by DFT. We calculate a rough version of coordinates by rdkit [18] for the validation set and the test set. The objective is predicting the HOMO-LUMO gap. The results are presented in Table 3 in terms of Mean Absolute Error (MAE) in eV unit. We compare our algorithm with 14 methods: GINE-VN [2, 6], GCN-VN [17, 6], GIN-VN [38, 6], DeeperGCN-VN [20, 6], TokenGT [16], GRPE [27], Graphormer [40], GraphGPS [28], GEM-2 [22], Transformer-M [24], GPS++ [25], Uni-Mol+ [23], and TGT-At [13].

As shown in Table 3, our method achieves competitive performance with other state-of-the-art methods. Though the MAE of our method on pre-training is slightly higher than that of TGT-At, the finetuning result in Sec. 5.3 is better, which shows greater potential on downstream tasks. In addition, the interpretability of our method is stronger, as will be specified in Sec. 5.4. The time cost of our method is competitive to other state-of-the-art methods with available efficiency data [23, 13], as shown in Table 4.

5.3 Finetuning

As the 3D coordinates are not provided for *PCBA*, we calculate coordinates by rdkit as well. The results are presented in Table 5 in terms of Average Precision (AP). We compare our algorithm with 7 methods: DeeperGCN [20], DGN [1], GINE [2], PHC-GNN [19], GIN-VN [38, 6], Graphormer [40] and TGT-At [13]. Our method outperforms the state-of-the-art approaches and achieves performance gain by around

Table 3: Results on *PCQM4Mv2*. The evaluation metric is the Mean Absolute Error (MAE↓) [eV]. Bold values indicate the best performance.

Model	Year	Source	Valid MAE↓	Test-dev MAE↓
GINE-VN [2, 6]	2020	arXiv preprint, ICML	0.1167	-
GCN-VN [17, 6]	2017	ICLR, ICML	0.1153	0.1152
GIN-VN [38, 6]	2018	ICLR, ICML	0.1083	0.1084
DeeperGCN-VN [20, 6]	2020	arXiv preprint, ICML	0.1021	-
TokenGT [16]	2022	NeurIPS	0.0910	0.0919
GRPE [27]	2022	ICLR	0.0867	0.0876
Graphormer [40]	2021	NeurIPS	0.0864	-
GraphGPS [28]	2022	NeurIPS	0.0852	0.0862
GEM-2 [22]	2022	arXiv preprint	0.0793	0.0806
Transformer-M [24]	2022	ICLR	0.0772	0.0782
GPS++ [25]	2022	arXiv preprint	0.0778	0.0720
Uni-Mol+ [23]	2023	Nature Communications	0.0693	0.0705
TGT-At [13]	2024	ICML	**0.0671**	**0.0683**
HieGT			0.0769	0.0781

4.99%, which gives credits to the effective local-global representation learning by the proposed hierarchical framework.

5.4 Interpretation

To show the interpretability of the proposed method, we visualize the attention scores of two randomly sampled molecules in head 0 and head 1 after AGA and MGA, compared with the state-of-the-art method TGT [13], as shown in Fig. 5. The results illustrate the main impact within intra-motif edges in AGA and inter-motif edges in MGA, which demonstrate the intuitive interpretability of the proposed method. For instance, in AGA, the attention weights are constrained in atoms of the same motif of the target atom. In MGA, the attention weights are significant in atoms of other motifs. In comparison, the distribution

Table 4: Efficiency comparison on *PCQM4Mv2*.

Model	GPUs	Training time	Inference time
Uni-Mol+ [23]	8 A100 GPUs	5 days	7 minutes
TGT [13]	8 A100 GPUs	4 days	-
HieGT	8 RTX3090 GPUs	5 days	8 minutes

Table 5: Results on *PCBA*. The evaluation metric is the Average Precision (AP↑). Bold values indicate the best performance.

Model	Year	Source	Test-AP(%)↑
DeeperGCN-VN-FLAG [20]	2020	arXiv preprint, ICML	28.42±0.43
DGN [1]	2021	ICML	28.85±0.30
GINE-VN [2, 6]	2020	arXiv preprint, ICML	29.17±0.15
PHC-GNN [19]	2021	ICANN	29.47±0.26
GIN-VN [38, 6]	2018	ICLR, ICML	29.02±0.17
Graphormer-FLAG [40]	2021	NeurIPS	31.40±0.34
TGT-Ag+TGT-At-DP [13]	2024	ICML	31.67±0.31
HieGT			**33.25**±0.27

of attention weights in TGT is spread around and it is not easy to discover chemical mechanism. Moreover, the atoms with significant weights are consistent in Head 0 and Head 1 in AGA (the neighbor atoms of the candidate atom) or MGA (atoms in other motifs), while atoms with significant weights vary greatly in Head 0 and Head 1 in TGT (either the neighbor atom of the candidate atom, or one of the farthest atoms). That is to say, the weights of atoms are more stable across different attention heads in our method. This indicates that the latent features learned by AGA and MGA are more intrinsic than TGT for molecular representation learning.

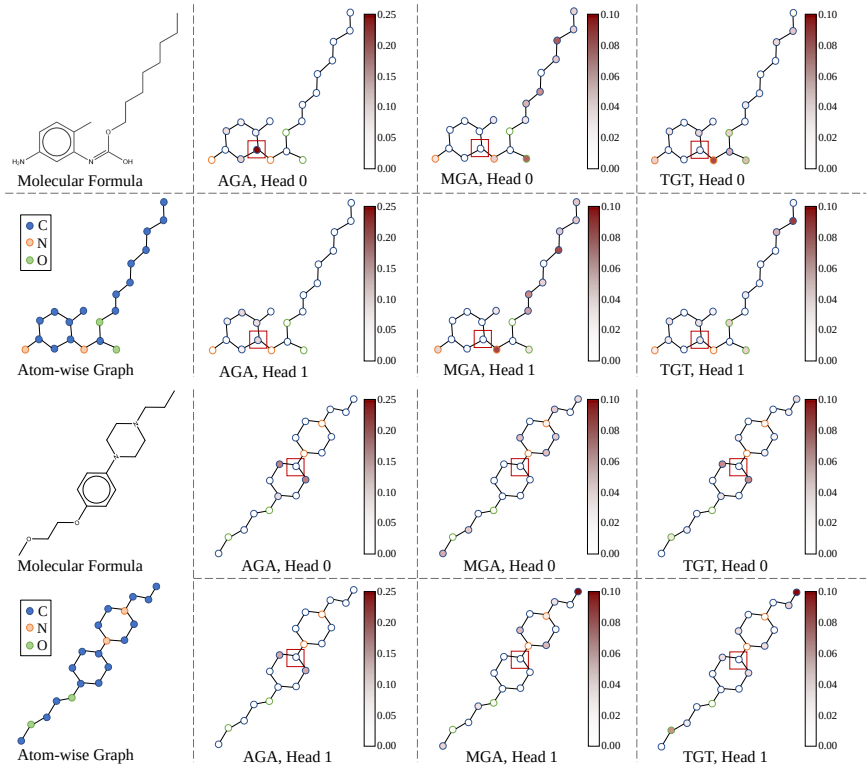

Figure 5: The visualization of attention weights of two randomly sampled molecules. The node marked by a red square represents the candidate atom, while the other atoms are shaded from red to white based on the magnitude of their attention weights relative to the candidate atom.

5.5 Ablation Study

We conduct ablation studies on the two major components of our algorithm: AGA and MGA. Experiments are conducted on the *PCQM4Mv2* dataset in Table 6. The evaluation metric is the Mean Absolute Error (MAE) in eV unit. The results with both AGA and MGA are significantly better than results with removing one of the modules, which validates the effectiveness of our method. In addition, keeping merely the AGA module achieves lower MAE than remaining merely MGA. This could suggest that atom features serve as a comparatively more important role in molecular representation learning.

Table 6: Ablation Study on *PCQM4Mv2*.

Model	AGA	MGA	Valid MAE↓
HieGT	✗	✓	0.0856
HieGT	✓	✗	0.0812
HieGT	✓	✓	**0.0769**

6 Conclusion

In this work, we propose Hierarchical Graph Transformer to learn local-global molecular representations. As molecules can be decomposed into motifs that possess local substructures and functions, we develop rules to construct motif-wise graphs from atom-wise graphs, and design Atom-wise Graph Attention and Motif-wise Graph Attention constrained by intra-motif edges and inter-motif edges. Experimental results show that the proposed method achieves competitive results on pretraining and significantly outperforms state-of-the-art graph representation learning approaches on finetuning. In future, we plan to apply the proposed method on more downstream tasks, such as molecule generation and molecular conformation learning.

References

[1] D. Beaini, S. Passaro, V. Létourneau, W. Hamilton, G. Corso, and P. Liò, "Directional graph networks", in *International Conference on Machine Learning*, PMLR, 2021, 748–58.

[2] R. Brossard, O. Frigo, and D. Dehaene, "Graph convolutions that can finally model local structure", *arXiv preprint arXiv:2011.15069*, 2020.

[3] K. Burke, "Perspective on density functional theory", *The Journal of chemical physics*, 136(15), 2012.

[4] A. Crum-Brown and T. R. Fraser, "The connection of chemical constitution and physiological action", *Trans R Soc Edinb*, 25(1968-1969), 1865, 257.

[5] Y. Fang, Q. Zhang, H. Yang, X. Zhuang, S. Deng, W. Zhang, M. Qin, Z. Chen, X. Fan, and H. Chen, "Molecular contrastive learning with chemical element knowledge graph", in *Proceedings*

of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 4, 2022, 3968–76.

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry", in *International conference on machine learning*, PMLR, 2017, 1263–72.

[7] Z. Guo, K. Guo, B. Nan, Y. Tian, R. G. Iyer, Y. Ma, O. Wiest, X. Zhang, W. Wang, C. Zhang, *et al.*, "Graph-based molecular representation learning", *arXiv preprint arXiv:2207.04869*, 2022.

[8] C. Hansch and T. Fujita, "p-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure", *Journal of the American Chemical Society*, 86(8), 1964, 1616–26.

[9] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)", *arXiv preprint arXiv:1606.08415*, 2016.

[10] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec, "Ogb-lsc: A large-scale challenge for machine learning on graphs", *arXiv preprint arXiv:2103.09430*, 2021.

[11] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs", *Advances in neural information processing systems*, 33, 2020, 22118–33.

[12] J. Huang, C. Sun, M. Li, R. Tang, B. Xie, S. Wang, and J.-M. Wei, "Structure-inclusive similarity based directed GNN: a method that can control information flow to predict drug–target binding affinity", *Bioinformatics*, 40(10), 2024, btae563.

[13] M. S. Hussain, M. J. Zaki, and D. Subramanian, "Triplet Interaction Improves Graph Transformers: Accurate Molecular Graph Learning with Triplet Graph Transformers", *arXiv preprint arXiv:2402.04538*, 2024.

[14] M. S. Hussain, M. J. Zaki, and D. Subramanian, "Global Self-Attention as a Replacement for Graph Convolution", in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, *KDD '22*, Washington DC, USA: Association for Computing Machinery, 2022, 655–65, ISBN: 9781450393850, DOI: 10.1145/3534678.3539296, https://doi.org/10.1145/3534678.3539296.

[15] W. Jin, R. Barzilay, and T. Jaakkola, "Junction Tree Variational Autoencoder for Molecular Graph Generation", in *Proceedings of the 35th International Conference on Machine Learning*, ed.

J. Dy and A. Krause, Vol. 80, *Proceedings of Machine Learning Research*, PMLR, October 2018, 2323–32, https://proceedings.mlr.press/v80/jin18a.html.

[16] J. Kim, D. Nguyen, S. Min, S. Cho, M. Lee, H. Lee, and S. Hong, "Pure transformers are powerful graph learners", *Advances in Neural Information Processing Systems*, 35, 2022, 14582–95.

[17] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks", 2016.

[18] G. Landrum, "Rdkit documentation", *Release*, 1(1-79), 2013, 4.

[19] T. Le, M. Bertolini, F. Noé, and D.-A. Clevert, "Parameterized hypercomplex graph neural networks for graph classification", in *International Conference on Artificial Neural Networks*, Springer, 2021, 204–16.

[20] G. Li, C. Xiong, A. Thabet, and B. Ghanem, "Deepergcn: All you need to train deeper gcns", *arXiv preprint arXiv:2006.07739*, 2020.

[21] S. Li, J. Zhou, T. Xu, D. Dou, and H. Xiong, "Geomgcl: Geometric graph contrastive learning for molecular property prediction", in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36, No. 4, 2022, 4541–9.

[22] L. Liu, D. He, X. Fang, S. Zhang, F. Wang, J. He, and H. Wu, "Gem-2: Next generation molecular property prediction network with many-body and full-range interaction modeling", *arXiv preprint arXiv:2208.05863*, 2022.

[23] S. Lu, Z. Gao, D. He, L. Zhang, and G. Ke, "Data-driven quantum chemical property prediction leveraging 3d conformations with uni-mol+", *Nature communications*, 15(1), 2024, 7104.

[24] S. Luo, T. Chen, Y. Xu, S. Zheng, T.-Y. Liu, L. Wang, and D. He, "One transformer can understand both 2d & 3d molecular data", *arXiv preprint arXiv:2210.01765*, 2022.

[25] D. Masters, J. Dean, K. Klaser, Z. Li, S. Maddrell-Mander, A. Sanders, H. Helal, D. Beker, L. Rampášek, and D. Beaini, "Gps++: An optimised hybrid mpnn/transformer for molecular property prediction", *arXiv preprint arXiv:2212.02229*, 2022.

[26] M. Nakata and T. Shimazaki, "PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry", *Journal of chemical information and modeling*, 57(6), 2017, 1300–8.

[27] W. Park, W. Chang, D. Lee, J. Kim, and S.-w. Hwang, "Grpe: Relative positional encoding for graph transformer", *arXiv preprint arXiv:2201.12787*, 2022.

[28] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a general, powerful, scalable graph transformer", *Advances in Neural Information Processing Systems*, 35, 2022, 14501–15.

[29] D. Rogers and M. Hahn, "Extended-connectivity fingerprints", *Journal of chemical information and modeling*, 50(5), 2010, 742–54.

[30] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers", *IEEE transactions on Signal Processing*, 45(11), 1997, 2758–65.

[31] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee, "Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction", *ACS central science*, 5(9), 2019, 1572–83.

[32] X. Tong, X. Liu, X. Tan, X. Li, J. Jiang, Z. Xiong, T. Xu, H. Jiang, N. Qiao, and M. Zheng, "Generative models for de novo drug design", *Journal of Medicinal Chemistry*, 64(19), 2021, 14011–27.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", *Advances in neural information processing systems*, 30, 2017.

[34] Y. Wang, S. Chen, G. Chen, E. Shurberg, H. Liu, and P. Hong, "Motif-Based Graph Representation Learning with Application to Chemical Molecules", in *Informatics*, Vol. 10, No. 1, MDPI, 2023, 8.

[35] D. Weininger, A. Weininger, and J. L. Weininger, "SMILES. 2. Algorithm for generation of unique SMILES notation", *Journal of chemical information and computer sciences*, 29(2), 1989, 97–101.

[36] M. Wen, E. W. C. Spotte-Smith, S. M. Blau, M. J. Mcdermott, A. S. Krishnapriyan, and K. A. Persson, "Chemical reaction networks and opportunities for machine learning", *Nature Computational Science*,

[37] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: a benchmark for molecular machine learning", *Chemical science*, 9(2), 2018, 513–30.

[38] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?", *arXiv preprint arXiv:1810.00826*, 2018.

[39] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, *et al.*, "Analyzing learned molecular representations for property prediction", *Journal of chemical information and modeling*, 59(8), 2019, 3370–88.

[40] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?", *Advances in Neural Information Processing Systems*, 34, 2021, 28877–88.

[41] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?", *Advances in neural information processing systems*, 34, 2021, 28877–88.

[42] Y. Zhang, Z. Deng, X. Xu, Y. Feng, and S. Junliang, "Application of Artificial Intelligence in Drug–Drug Interactions Prediction: A Review", *Journal of Chemical Information and Modeling*, 2023.

[43] Y. Zhang, Z. Deng, X. Xu, Y. Feng, and S. Junliang, "Application of Artificial Intelligence in Drug–Drug Interactions Prediction: A Review", *Journal of chemical information and modeling*, 64(7), 2023, 2158–73.

[44] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C.-K. Lee, "Motif-based graph self-supervised learning for molecular property prediction", *Advances in Neural Information Processing Systems*, 34, 2021, 15870–82.