# Association Pattern-enhanced Molecular Representation Learning

**Lingxiang Jia**[1,2]**, Yuchen Ying**[1,2]**, Tian Qiu**[1,2]**, Shaolun Yao**[1,2]**,**
**Liang Xue**[3*]**, Jie Lei**[4]**, Jie Song**[1,2]**, Mingli Song**[1,2]**, Zunlei Feng**[1,2*]

[1]State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3]Computing Science and Artificial Intelligence College, Suzhou City University
[4]College of Computer Science, Zhejiang University of Technology
{lingxiangjia, yingyc, tqiu, yaoshaolun}@zju.edu.cn, xliang@szcu.edu.cn,
jasonlei@zjut.edu.cn, {sjie, brooksong, zunleifeng}@zju.edu.cn

## Abstract

The applicability of drug molecules in various clinical scenarios is significantly influenced by a diverse range of molecular properties. By leveraging self-supervised conditions such as atom attributes and interatomic bonds, existing advanced molecular foundation models can generate expressive representations of these molecules. However, such models often overlook the fixed association patterns within molecules that influence physiological or chemical properties. In this paper, we introduce a novel association pattern-aware message passing method, which can serve as an effective yet general plug-and-play plugin, thereby enhancing the atom representations generated by molecular foundation models without requiring additional pretraining. Additionally, molecular property-specific pattern libraries are constructed to collect the generated interpretable common patterns that bind to these properties. Extensive experiments conducted on 11 benchmark molecular property prediction tasks across 8 advanced molecular foundation models demonstrate significant superiority of the proposed method, with performance improvements of up to approximately 20%. Furthermore, a property-specific pattern library is tailored for blood-brain barrier penetration, which has undergone corresponding mechanistic validation.

**Code & Appendix** — https://github.com/hry98kki/APMP

## Introduction

Molecular properties, encompassing aspects such as physiology, biophysics, and physical chemistry, play a critical role in determining the viability and usability of pharmaceutical compounds (Glaser 2012; Widmaier, Raff, and Strang 2022; Silbey et al. 2022). These properties often pose significant challenges during clinical trials. For instance, extensive toxicity testing is typically required, wherein medical and pharmaceutical experts evaluate the potential presence of toxic structural alerts in drug candidates. This process is followed by in vivo and in vitro experiments to further assess the drug's safety and feasibility. While such thorough evaluation is essential to ensure patient safety, it also substantially increases time and resource consumption associated with drug development (DiMasi, Grabowski, and Hansen 2016). Thus, efficient strategies are required to promote drug discovery.
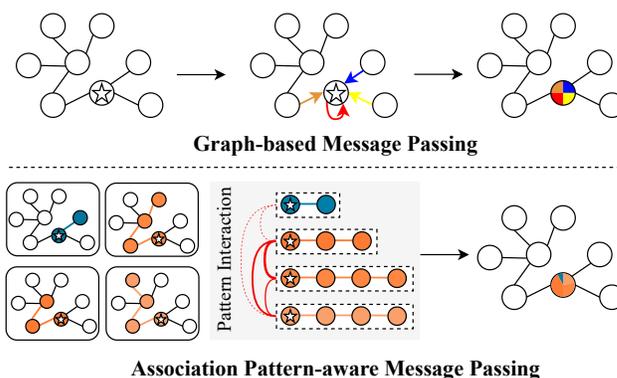
Figure 1: Illustration of message passing of graph-based and association pattern-aware for atom representation updates.

The rapid development of deep learning has significantly advanced the automation of molecular property prediction for drug compounds. Given the inherent graph structure of drug molecules, the encoding process for molecular graphs is typically designed as message passing based on graph neural networks (Kipf and Welling 2016; Veličković et al. 2018; Xu et al. 2019; Yao et al. 2022), which aggregate information from neighboring nodes to update the representation of a central atom, focusing on local contexts, as illustrated in Figure 1. Currently, state-of-the-art methods in this field predominantly rely on molecular foundation models (MFMs) (Xia et al. 2023). These models utilize various self-supervised signals, such as atom attributes, interatomic bonds, and 3D structures, to effectively capture complex relationships within molecular structures and expressive representations of molecular properties through pretraining on vast datasets of unlabeled molecular data using graph-based encoders, thus allowing them to transfer the learned knowledge to a wide range of downstream property prediction tasks, such as predicting toxicity, solubility, and other physicochemical properties of these drug candidates.

However, the aforementioned architectures typically overlook the impact of path patterns on learning of expressive atom representations, even though these patterns are directly relevant to target properties (Nelson 1982; Al-Hasani and

Bruchas 2011; Kalgutkar 2019; Jia et al. 2022; Yang et al. 2022), limiting the ability to fully capture complex atomic interactions within molecular graphs. For instance, common patterns linking the phenyl and amino groups, known as arylamines, typically determine the liver toxicity in humans (Smith et al. 2003; Usui et al. 2009). Therefore, specific patterns associated with molecular properties should be considered during chemical message passing through atoms in the graph. By extracting such association path-aware information, atomic representations—and consequently, the entire molecular graph—become more comprehensive and meaningful due to the interpretability of common patterns related to specific properties.

To enhance atom representations from the perspective of association patterns, we propose a novel association path-aware message passing method that explicitly integrates relevant patterns within molecular structures. The proposed method can be seamlessly integrated into existing MFMs as a plugin to enhance the representation learning process. Specifically, by exploring underlying path patterns, the proposed method generates expressive pattern-aware representations for molecular properties, forming an interpretable library of property-specific patterns. Experimental results on property prediction tasks demonstrate the effectiveness of the proposed association path-aware message passing.

Our contributions are summarized as follows:

- We propose a novel association path-aware message passing method that samples high-confidence biological patterns associated with nodes and thus extracts common patterns within molecular graphs, which can effectively enhance the expressive ability of atom representations.

- A simple yet general association pattern-enhanced plugin is designed, which can be easily integrated into existing MFMs without requiring a complex and time-consuming pretraining process.

- By applying the pattern-enhanced plugin to 8 advanced MFMs, the performance improvements achieve up to 20% across 11 benchmark property prediction tasks.

- An interpretable library of high-confidence patterns is constructed for blood-brain barrier penetration, thereby offering valuable insights for drug discovery.

## Related Work

**Molecular Foundation Models.** Inspired by the success of large language models, pretrained MFMs have been developed to learn universal molecular representations from massive unlabeled molecules and then finetuned on specific downstream tasks (Xia et al. 2023). N-GRAM (Liu, Demirel, and Liang 2019) assembled atom node embeddings from short walks and employed simple and classical methods to predict molecular properties. GROVER (Rong et al. 2020) integrated both atom-level and graph-level knowledge through pretext tasks. Building on atom-level information, MGSSL (Zhang et al. 2021) incorporated a motif generation supervised task. GEM (Fang et al. 2022) and Graph-MVP (Liu et al. 2022) utilized 2D topological structures and 3D geometric views to generate comprehensive molecular representations, while UniMol (Zhou et al. 2023) em-

ployed 3D molecular conformations to perform pretraining tasks involving 3D position recovery and masked atom prediction. Moreover, MolCLR (Wang et al. 2022) introduced a contrastive learning framework by applying general graph augmentation techniques to approximately 10 million unique molecules, including atom masking, bond deletion, and subgraph removal. Despite their successful applications, these advanced methods assume that atom representations are learned through interactions among neighboring atoms or predefined molecular substructures, resulting in a lack of specificity regarding properties, which is insufficient for comprehensively learning complex interactions for a wide range of molecular properties.

**Path-based Graph Mining.** Path-based graph mining techniques have been widely employed to extract local structural information from networks. For homogeneous networks, random walk-based strategy are usually utilized to mine the graph information. Brin and Page (1998) proposed a classic ranking algorithm PageRank based on the link structure between web pages to obtain the importance score of each page, and Jeh and Widom (2003) further proposed a personalized extension for each node. Jeh and Widom (2002) adopted a similarity measure based on pairwise random walk, which can capture the structural similarity between nodes. Perozzi, Al-Rfou, and Skiena (2014) proposed a deep learning method that leverages local random walk information to learn vertex latent representations.

For heterogeneous networks, meta-path-based methods have been introduced to capture contextual information within the network. Sun et al. (2011) proposed a meta-path-based similarity framework that captures subtle semantic similarities among objects of the same type in the network. Dong, Chawla, and Swami (2017) proposed a deep learning-based heterogeneous network representation learning method that automatically learns hidden meta-path semantics, generating general node embedding representations. Wang et al. (2023) addressed the issue of low homophily in graph-based fraud detection by integrating label information to generate distinguishable neighborhood information. Specifically, for hypergraph networks, Tu et al. (2018) proposed a deep hyper-network embedding model to preserve both local and global proximities in the embedding space. Zhang, Zou, and Ma (2020) developed a self-attention-based graph neural network applicable to both homogeneous and heterogeneous hypergraphs. Jia et al. (2024) introduced an interpretable rule-mining method to enhance node representations, despite the fact that these rules are constrained by fixed lengths and lack a clear mechanism.

## The Proposed Method

To expand the expressive capability of atom representations, we propose a novel association pattern-aware message passing method. The method is further designed as a simple yet general plugin that can be easily integrated into existing MFMs. Additionally, the molecular property-specific pattern libraries—containing common biological pathways—are constructed based on the pattern-enhanced module. See Figure 2 for the illustrations.
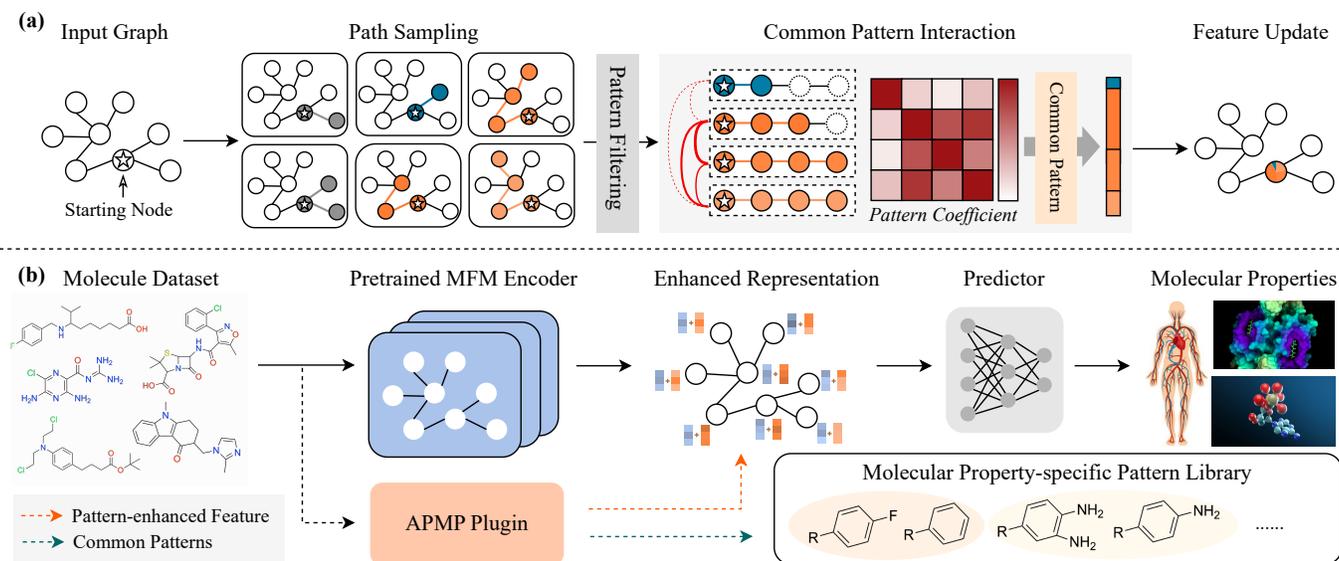
Figure 2: Illustrations of the proposed association pattern-enhanced method. (a) The association pattern-aware message passing process for certain node in the molecular graph. (b) The pipeline of the APMP method integrated into MFMs as a plugin.

Hence, the three parts of this section address the following research questions, respectively: **RQ1**: How to sample and fuse association patterns in molecular graphs? **RQ2**: Where to integrate the association pattern-enhanced module into the MFM architectures? **RQ3**: What to be derived from mining association patterns for molecular properties?

## Association Pattern-aware Message Passing

To comprehensively address feature propagation and interaction of association patterns, the proposed Association Pattern-aware Message Passing (APMP) module comprises three core components: First, the Pattern Sampling (PS) block is designed to sample potential patterns for each atom node within the molecular graph based on random walk. Next, the Confidence-based Pattern Filtering (CPF) mechanism is devised to filter out low-confidence patterns and retain the most representative ones. Finally, the Common Pattern Interaction (CPI) block is introduced to update node features through message interaction of the sampled association patterns and mine common patterns, thus solving **RQ1**.

**Pattern Sampling.** Considering the molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ presents the set of atoms, $\mathcal{E}$ presents the set of bonds. Technically, $\mathcal{G}$ is further formulated as an attributed graph with an adjacency matrix $\mathbf{H} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ and node attributes $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times f}$, which is defined as:

$$\mathbf{H}[u,v] = \begin{cases} 1, & \text{if node } u \text{ links to } v \\ 0, & \text{if node } u \text{ does not link to } v. \end{cases} \quad (1)$$

To mine potential patterns for each atom node, the random walk (Xia et al. 2019) is adopted to sample connected paths with variable length. Specifically, the random walk process starts from the target atom node $v$, constructing paths step by step by randomly selecting neighboring nodes.

Each step follows the node connectivity relationships from $\mathbf{H}$ to ensure that paths are validly generated within graph $\mathcal{G}$. The generation of paths ends with setting a step limit of $L$. Formally, this process can be described as $\mathbf{p}_v = \text{RandomWalk}(\mathcal{G}, v, L)$, where $\mathbf{p}_v \in \mathbb{Z}^L$ and $\mathbf{p}_v[i]$ denotes the $i$-th node starting from node $v$. Additionally, to expand the robustness and diversity of the generated path, the randomized sequential masking mechanism is adopted to randomly obscure the latter portion of paths with mask matrices $\mathbf{m} \in \{0,1\}^L$, which is formulated as:

$$\mathbf{p}_v = \mathbf{p}_v \odot (\mathbf{1}_L - \mathbf{m}) + (-\mathbf{1})_L \odot \mathbf{m}, \quad (2)$$

where $\odot$ denotes the element-wise multiplication operation. If $\mathbf{p}_v[i] = -1$, it indicates that the $i$-th node is masked.

Following the above process, $N$ different random paths are generated for each starting node $v$. These paths share the same starting node but may visit different neighboring nodes during the walk, producing multiple distinct paths. Consequently, the entire path set is denoted as $\mathbf{P}_v \in \mathbb{Z}^{N \times L}$, where $\mathbf{P}_v[j]$ denotes the $j$-th path starting from node $v$.

**Confidence-based Pattern Filtering.** After generating the $N$ random paths $\mathbf{P}_v$ for node $v$, they are evaluated and thus filtered to ensure the selection of top $n$ representative paths as the candidates of common pathway patterns.

**Definition 1** *Given certain sampled path $p_v$ starting from node $v$, the confidence of $p_v$ is defined as the cumulative sum of the structural attributes of nodes along $p_v$, representing the property-specific pathway contribution for node $v$.*

Following Definition 1, the centrality-based confidence scoring mechanism is introduced to assess the significance of each path by summing the centrality scores of all nodes along the path. Specifically, the degree centrality for node $v$ is the fraction of nodes it is connected to. The degree centrality values are normalized by dividing by the maximum

possible degree, i.e. $|\mathcal{V}| - 1$. The formula for the degree centrality $C_{\mathcal{G}}(v)$ of node $v$ is as follows:

$$C_{\mathcal{G}}(v) = \frac{1}{|\mathcal{V}| - 1} \sum_{u \in \mathcal{V}} \mathbf{H}[u, v]. \tag{3}$$

Subsequently, the path confidence score $\mathbf{S}_v$ is calculated based on the centrality of nodes within each path. The generated paths are then sorted according to $\mathbf{S}_v$, and the top $n$ paths with the highest scores are selected to form the final retained path set $\mathbf{P}_v^* \subseteq \mathbf{P}_v$. This process can be formulated as follows:

$$\mathbf{S}(p_v) = \sum_{v' \in p_v} C_{\mathcal{G}}(v'), \mathbf{P}_v^* = \arg\max(\mathbf{S}_v, \mathbf{P}_v, n). \tag{4}$$

Hence, the filtered path set $\mathbf{P}_v^* \in \mathbb{Z}^{n \times L}$ exhibits paths that are more effective and expressive for subsequent mining of common patterns.

Furthermore, the corresponding path feature $\mathbf{F}_v^* \in \mathbb{R}^{n \times L \times f}$ is denoted by the concatenation of nodes along each path with the $f$-dimension initial atom attributes. A linear projection layer then maps each path feature from $\mathbb{R}^{L \times f} \to \mathbb{R}^d$. Note that if there are empty nodes in the path, the zero-padding is applied to fill the path feature.

**Common Pattern Interaction.** Given the feature vector $\mathbf{F}_v^*$ of $n$ high-confidence paths for node $v$, the goal is to mine the common pattern inside these paths. To this end, the CPI block is designed to aggregate the information of these paths, which consists of a composition of Transformer layers (Vaswani et al. 2017). Each Transformer layer has two parts: a multi-head attention module (MHA) and a position-wise feed-forward network (FFN). Let $\mathbf{z}_v = \mathbf{F}_v^* \in \mathbb{R}^{n \times d}$ denotes the input of the MHA, which is projected by three matrices $\mathbf{W}_Q \in \mathbb{R}^{d \times d_K}$, $\mathbf{W}_K \in \mathbb{R}^{d \times d_K}$, and $\mathbf{W}_V \in \mathbb{R}^{d \times d_V}$ to obtain the corresponding query, key, and value representations $\mathbf{Q}_v$, $\mathbf{K}_v$, and $\mathbf{V}_v$ for node $v$:

$$\mathbf{Q}_v = \mathbf{z}_v \mathbf{W}_Q, \mathbf{K}_v = \mathbf{z}_v \mathbf{W}_K, \mathbf{V}_v = \mathbf{z}_v \mathbf{W}_V, \tag{5}$$

$$\mathbf{A}_v = \frac{\mathbf{Q}_v \mathbf{K}_v^\top}{\sqrt{d_K}}, \mathrm{Attn}(\mathbf{z}_v) = \mathrm{softmax}(\mathbf{A}_v)\mathbf{V}_v, \tag{6}$$

where $\mathbf{A}_v \in \mathbb{R}^{n \times n}$ is a matrix capturing the similarity between queries and keys, which denotes the quantitative relation of $n$ paths. For simplicity, we consider the single-head attention and assume $d_K = d_V = d$. The extension to the multi-head attention is standard and straightforward. Then we will get the output of the MHA module $\mathbf{z}_v' \in \mathbb{R}^{n \times d}$.

To summarize the process of the CPI block for the entire graph $\mathcal{G}$, let $\mathbf{Z}^{(0)} = [\mathbf{z}_1^{(0)\top}, \mathbf{z}_2^{(0)\top}, \ldots, \mathbf{z}_{|\mathcal{V}|}^{(0)\top}]^\top \in \mathbb{R}^{|\mathcal{V}| \times n \times d}$ is denoted as the input embeddings of $\mathcal{G}$, then the feature aggregation in these transformer layers is computed as:

$$\mathbf{Z}'^{(t)} = \mathrm{MHA}(\mathrm{LN}(\mathbf{Z}^{(t-1)})) + \mathbf{Z}^{(t-1)}, \tag{7}$$

$$\mathbf{Z}^{(t+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Z}'^{(t)}) + \mathbf{Z}'^{(t)}, \tag{8}$$

where LN is the layer normalization, and $\mathbf{Z}^{(t+1)}$ is the output of the current transformer layer and the input of the

---

**Algorithm 1: Pattern-Enhanced Finetuning for MFMs**

**Input**: Molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with features $\mathbf{X}$, target property label $y$
**Parameters**: MFM encoder $\Theta_{\mathrm{MFM}}$; APMP module $\Theta_{\mathrm{APMP}}$; Predictor $\Theta_{\mathrm{Pred}}$; Learning rate for three modules $\alpha_1, \alpha_2, \alpha_3$
**Output**: Predicted property label $\hat{y}$

1: **while** not converged **do**
2:     $\mathcal{L} \leftarrow 0$ {Initialize loss for the batch}
3:     **for** each molecular graph $\mathcal{G}_k$ in the batch **do**
4:         **for** $v_i \in \mathcal{V}_k$ **do**
5:             $\mathbf{z}_{v_i} \leftarrow \mathrm{APMP}(v_i, \mathbf{X}_k; \Theta_{\mathrm{APMP}})$
6:         **end for**
7:         $\mathbf{M}_k \leftarrow \mathrm{MFM\text{-}Encoder}(\mathbf{X}_k; \Theta_{\mathrm{MFM}})$
8:         **for** $v_i \in \mathcal{V}_k$ **do**
9:             $\mathbf{h}_{v_i} \leftarrow \mathbf{z}_{v_i} + \mathbf{M}_{v_i}$ {Element-wise addition}
10:         **end for**
11:         $\mathbf{g}_k \leftarrow \mathrm{GlobalPooling}(\{\mathbf{h}_{v_i} | v_i \in \mathcal{V}_k\})$
12:         $\hat{y}_k \leftarrow \mathrm{Predictor}(\mathbf{g}_k; \Theta_{\mathrm{Pred}})$
13:         $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{Loss}(\hat{y}_k, y_k)$
14:     **end for**
15:     $\mathcal{L} \leftarrow \mathcal{L}/\mathrm{BatchSize}$
16:     $\Theta_{\mathrm{MFM}} \leftarrow \Theta_{\mathrm{MFM}} - \alpha_1 \frac{\partial \mathcal{L}}{\partial \Theta_{\mathrm{MFM}}}$
17:     $\Theta_{\mathrm{APMP}} \leftarrow \Theta_{\mathrm{APMP}} - \alpha_2 \frac{\partial \mathcal{L}}{\partial \Theta_{\mathrm{APMP}}}$
18:     $\Theta_{\mathrm{Pred}} \leftarrow \Theta_{\mathrm{Pred}} - \alpha_3 \frac{\partial \mathcal{L}}{\partial \Theta_{\mathrm{Pred}}}$
19: **end while**
20: **return** $\hat{y}$

---

next layer. Thus, after $T$ layers, we get the encoding output $\mathbf{Z}^{(T)} = [\mathbf{z}_1^{(T)\top}, \mathbf{z}_2^{(T)\top}, \ldots, \mathbf{z}_{|\mathcal{V}|}^{(T)\top}]^\top \in \mathbb{R}^{|\mathcal{V}| \times n \times d}$, and then apply the mean function to fuse $n$ association patterns for node $v$ to get the learned association pattern-aware embedding $\mathbf{z}_v \in \mathbb{R}^d$. Similarly, the above strategy is applied to obtain embeddings for all other nodes within the graph.

## Dual-branch Pattern-enhanced Representation

To effectively supplement expressive representations from the association pattern perspective, the dual-branch pattern-enhanced finetune strategy is proposed to enhance the atom representation generated by MFMs, thus addressing **RQ2**.

Considering the variable designs of MFMs, the APMP module can serve as a dual-branch plugin independent of the MFM main encoder, allowing appropriate finetuning for the downstream task prediction. This enhancement strategy effectively boosts the representation without requiring in-depth model redesign, thus achieving generalizability and eliminating the need for extensive retraining. Specifically, during the finetune process, the pattern representation $\mathbf{z}_v$ of node $v$ obtained from the pattern-enhanced module is integrated into the atom representation $\mathbf{M}_v$ obtained from the pretrained encoders of MFMs. Global pooling function is then performed on the enhanced atom representations to obtain the graph representation, which is fed into the predictor to predict the final property label. Figure 2b shows an illustration, and the whole finetune process is formulated in Algorithm 1. Furthermore, the in-depth validation of this plugin is provided through Table 3 in ablation experiments.

## Molecular Property-specific Pattern Library

To mine the intrinsic mechanism of the common patterns for specific molecular property, the contribution-based pattern scoring method is proposed to utilize the contribution of common patterns to atoms based on the trained APMP module and construct the property-specific pattern libraries, thereby addressing **RQ3**.

Specifically, the relationships of these high-confidence patterns are defined in Equation 6, i.e. $\mathbf{A}_v \in \mathbb{R}^{n \times n}$, where $n$ denotes the number of high-confidence patterns after the CPF block. Then, the contributions of these patterns to node $v$ are calculated by a function, termed by *Common Pattern Coefficient* (CPC). The contribution of $n$ patterns to node $v$ calculated through CPC function is formulated as follows:

$$\tilde{\mathbf{A}}_v = \text{softmax}(\mathbf{A}_v), \tag{9}$$

$$\mathbf{CPC}(v) = \frac{1}{n} \sum_{j=1}^{n} \tilde{\mathbf{A}}_v[j,:]. \tag{10}$$

Based on the CPC function, the contribution ranking for $n$ patterns associated with node $v$ can be conducted. Patterns with relatively high contribution values are identified and compiled into a set of common pattern candidates. After all common patterns across the molecular graph are retrieved, a filtering process is applied to obtain a molecular property-specific pattern library. Furthermore, the pattern library is comprehensively validated through the biological mechanisms referenced in the corresponding literature.

## Complexity Analysis

Considering the APMP, the critical CPI module primary including several MHA layers and FFN layers. For certain atom node with $n$ high-confidence patterns obtained from the CPF block, the dimension of the input and hidden features for the MHA layer are assumed $d$, and the dimension of hidden features for the FFN layer is $d'$. The query, key, and value matrices are derived from the same input sequence and share the same length $n$. In CPI, the primary operations include scaled dot-product attention, multiplication of attention weights and values, MHA linear transformation, and FFN linear projection. The time complexity is expressed as $\mathcal{O}(n^2 \cdot d + n \cdot d^2 + n \cdot d \cdot d')$. Hence, for the entire molecular graph $\mathcal{G}$, the total complexity is $\mathcal{O}\left(|\mathcal{V}| \cdot (n^2 \cdot d + n \cdot d^2 + n \cdot d \cdot d')\right)$, where $|\mathcal{V}|$ is the number of atom nodes in the $\mathcal{G}$.

# Experiments

## Experimental Setting

**Datasets.** To comprehensively evaluate the proposed method on downstream tasks, we conduct experiments on 11 benchmark datasets from MoleculeNet (Wu et al. 2018), covering diverse targets, including physical chemistry (ESOL, FreeSolv, Lipophilicity), biophysics (BACE, MUV, HIV), and physiology (BBBP, Tox21, ToxCast, SIDER, ClinTox). Among these tasks, ESOL, FreeSolv, and Lipophilicity are formulated as regression tasks, while the others are treated as classification tasks. Details of these datasets can be found at Appendix C.1.

| Regression (RMSE, lower is better ↓) | | | |
|---|---|---|---|
| Datasets | ESOL | FreeSolv | Lipophilicity |
| # Molecules | 1,128 | 642 | 4,200 |
| N-GRAM | $0.8290_{\pm 0.0197}$ | $0.4609_{\pm 0.1251}$ | $0.8090_{\pm 0.0091}$ |
| + *APMP* | $0.8153_{\pm 0.0188}$ | $0.4524_{\pm 0.0897}$ | $0.7940_{\pm 0.0248}$ |
| Impr. | **+1.65%** | **+1.84%** | **+1.85%** |
| GROVER | $0.6239_{\pm 0.0471}$ | $1.4754_{\pm 0.2627}$ | $0.7421_{\pm 0.0341}$ |
| + *APMP* | $0.6073_{\pm 0.0309}$ | $1.1922_{\pm 0.2922}$ | $0.6804_{\pm 0.0733}$ |
| Impr. | **+2.67%** | **+19.20%** | **+8.31%** |
| MGSSL | $1.0792_{\pm 0.0216}$ | $2.4261_{\pm 0.0346}$ | $0.8726_{\pm 0.0079}$ |
| + *APMP* | $1.0701_{\pm 0.0110}$ | $2.3427_{\pm 0.0738}$ | $0.8673_{\pm 0.0095}$ |
| Impr. | **+0.84%** | **+3.44%** | **+0.61%** |
| GEM | $0.4668_{\pm 0.1277}$ | $0.8810_{\pm 0.3863}$ | $0.4685_{\pm 0.0671}$ |
| + *APMP* | $0.4606_{\pm 0.0346}$ | $0.8211_{\pm 0.3856}$ | $0.4461_{\pm 0.0579}$ |
| Impr. | **+1.33%** | **+6.80%** | **+4.78%** |
| GraphMVP | $0.6535_{\pm 0.0399}$ | $1.6031_{\pm 0.3622}$ | $0.6072_{\pm 0.0610}$ |
| + *APMP* | $0.6532_{\pm 0.0371}$ | $1.5688_{\pm 0.3991}$ | $0.6011_{\pm 0.0669}$ |
| Impr. | **+0.05%** | **+2.14%** | **+1.01%** |
| MolCLR$_{GCN}$ | $1.1428_{\pm 0.1512}$ | $2.4033_{\pm 0.0983}$ | $0.7621_{\pm 0.0472}$ |
| + *APMP* | $1.1138_{\pm 0.1581}$ | $2.3037_{\pm 0.2192}$ | $0.7584_{\pm 0.0452}$ |
| Impr. | **+2.54%** | **+4.14%** | **+0.49%** |
| MolCLR$_{GIN}$ | $0.8098_{\pm 0.0450}$ | $1.7434_{\pm 0.4078}$ | $0.6330_{\pm 0.0352}$ |
| + *APMP* | $0.8062_{\pm 0.0329}$ | $1.5701_{\pm 0.1459}$ | $0.6271_{\pm 0.0353}$ |
| Impr. | **+0.45%** | **+9.94%** | **+0.93%** |
| UniMol | $0.6322_{\pm 0.0106}$ | $0.7350_{\pm 0.0670}$ | $0.5571_{\pm 0.0015}$ |
| + *APMP* | $0.5867_{\pm 0.0429}$ | $0.7160_{\pm 0.1135}$ | $0.5552_{\pm 0.0072}$ |
| Impr. | **+7.20%** | **+2.59%** | **+0.34%** |

Table 1: Performance comparison on molecular property prediction regression tasks. All scores are represented as "mean$_{\pm \text{std}}$" with three different seeds. The third lines (Impr.) denote performance improvement with the *APMP* module.

**Baselines.** We comprehensively evaluate the proposed APMP plugin integrated into 8 advanced MFMs, including N-GRAM (Liu, Demirel, and Liang 2019), GROVER (Rong et al. 2020), MGSSL (Zhang et al. 2021), GEM (Fang et al. 2022), GraphMVP (Liu et al. 2022), MolCLR (Wang et al. 2022) and UniMol (Zhou et al. 2023). For MolCLR, the backbone utilizes both Graph Convolutional Networks (GCN) (Kipf and Welling 2016) and Graph Isomorphism Network (GIN) (Xu et al. 2019).

**Implementation Details.** To accurately evaluate model performance and prevent overfitting, three different seeds are used to evaluate the performance for each dataset and the mean and standard deviations are reported as the prediction results. Moreover, we randomly split all the datasets with a ratio for train/validation/test as 8:1:1, and use the best of validation results to select the best model evaluated on the test set. More finetuning details are found at Appendix C.2.

**Metrics.** Following the evaluation setting of MoleculeNet (Wu et al. 2018), we adopt the ROC-AUC score as the evaluation metric for eight molecular property classification tasks and RMSE for three molecular property regression tasks.

| Classification (ROC-AUC %, higher is better ↑) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Datasets | BBBP | BACE | ClinTox | Tox21 | ToxCast | SIDER | HIV | MUV |
| # Molecules | 2,039 | 1,513 | 1,478 | 7,831 | 8,575 | 1,427 | 41,127 | 93,087 |
| # Tasks | 1 | 1 | 2 | 12 | 617 | 27 | 1 | 17 |
| N-GRAM | $90.24_{\pm2.01}$ | $88.09_{\pm3.75}$ | $84.44_{\pm4.27}$ | $82.03_{\pm0.83}$ | - | - | - | - |
| + *APMP* | $91.92_{\pm1.75}$ | $88.65_{\pm3.63}$ | $86.83_{\pm3.22}$ | $82.91_{\pm1.99}$ | - | - | - | - |
| Impr. | **+1.68** | **+0.56** | **+2.39** | **+0.88** | - | - | - | - |
| GROVER | $89.42_{\pm1.70}$ | $83.13_{\pm1.24}$ | $79.93_{\pm3.07}$ | $78.40_{\pm1.34}$ | $75.06_{\pm1.17}$ | $65.93_{\pm1.77}$ | $80.58_{\pm2.25}$ | $78.02_{\pm2.19}$ |
| + *APMP* | $90.52_{\pm1.34}$ | $83.57_{\pm3.30}$ | $82.94_{\pm1.91}$ | $79.34_{\pm0.23}$ | $75.51_{\pm0.68}$ | $65.96_{\pm1.74}$ | $80.90_{\pm1.82}$ | $78.16_{\pm1.73}$ |
| Impr. | **+1.10** | **+0.44** | **+3.01** | **+0.94** | **+0.45** | **+0.03** | **+0.32** | **+0.14** |
| MGSSL | $91.55_{\pm0.18}$ | $87.88_{\pm0.28}$ | $77.94_{\pm1.12}$ | $80.06_{\pm0.18}$ | $70.15_{\pm0.07}$ | $62.70_{\pm0.49}$ | $84.32_{\pm0.40}$ | $82.07_{\pm0.37}$ |
| + *APMP* | $91.74_{\pm0.31}$ | $88.21_{\pm0.34}$ | $77.99_{\pm0.21}$ | $81.34_{\pm0.18}$ | $70.54_{\pm0.13}$ | $63.23_{\pm1.62}$ | $84.61_{\pm0.45}$ | $84.65_{\pm1.62}$ |
| Impr. | **+0.19** | **+0.33** | **+0.05** | **+1.28** | **+0.39** | **+0.53** | **+0.29** | **+2.58** |
| GEM | $92.99_{\pm1.06}$ | $88.77_{\pm1.38}$ | $91.38_{\pm0.89}$ | $85.58_{\pm0.72}$ | $74.83_{\pm1.04}$ | $67.04_{\pm0.57}$ | $82.78_{\pm2.85}$ | $77.46_{\pm0.41}$ |
| + *APMP* | $93.76_{\pm2.65}$ | $90.14_{\pm0.76}$ | $93.19_{\pm2.28}$ | $85.74_{\pm0.91}$ | $75.19_{\pm0.87}$ | $67.45_{\pm2.55}$ | $83.03_{\pm0.63}$ | $80.68_{\pm2.27}$ |
| Impr. | **+0.77** | **+1.37** | **+1.81** | **+0.16** | **+0.36** | **+0.41** | **+0.25** | **+3.22** |
| GraphMVP | $90.43_{\pm1.71}$ | $86.36_{\pm3.10}$ | $67.68_{\pm10.29}$ | $84.52_{\pm1.33}$ | $74.14_{\pm0.82}$ | $63.46_{\pm1.81}$ | $81.47_{\pm1.62}$ | $79.08_{\pm3.27}$ |
| + *APMP* | $91.11_{\pm1.71}$ | $86.76_{\pm3.10}$ | $68.92_{\pm13.98}$ | $84.59_{\pm0.74}$ | $74.14_{\pm0.55}$ | $63.50_{\pm0.52}$ | $82.00_{\pm1.15}$ | $79.62_{\pm0.99}$ |
| Impr. | **+0.68** | **+0.40** | **+1.24** | **+0.07** | **+0.00** | **+0.04** | **+0.53** | **+0.54** |
| MolCLR$_{GCN}$ | $90.64_{\pm0.97}$ | $80.25_{\pm2.34}$ | $79.90_{\pm3.47}$ | $81.49_{\pm0.62}$ | - | - | $82.19_{\pm0.60}$ | - |
| + *APMP* | $90.78_{\pm0.67}$ | $80.39_{\pm1.54}$ | $81.16_{\pm4.17}$ | $81.99_{\pm0.88}$ | - | - | $82.21_{\pm1.37}$ | - |
| Impr. | **+0.14** | **+0.14** | **+1.26** | **+0.50** | - | - | **+0.02** | - |
| MolCLR$_{GIN}$ | $92.71_{\pm0.42}$ | $85.77_{\pm2.59}$ | $87.07_{\pm3.81}$ | $82.59_{\pm0.60}$ | - | - | $83.22_{\pm1.47}$ | - |
| + *APMP* | $93.53_{\pm0.92}$ | $86.04_{\pm2.99}$ | $87.68_{\pm1.74}$ | $82.76_{\pm0.50}$ | - | - | $83.74_{\pm1.29}$ | - |
| Impr. | **+0.82** | **+0.27** | **+0.61** | **+0.19** | - | - | **+0.52** | - |
| UniMol | $89.22_{\pm1.58}$ | $87.79_{\pm3.09}$ | $89.35_{\pm3.07}$ | $84.90_{\pm0.30}$ | $71.43_{\pm1.40}$ | $63.62_{\pm3.22}$ | $82.95_{\pm0.56}$ | $75.84_{\pm2.02}$ |
| + *APMP* | $93.70_{\pm0.23}$ | $89.62_{\pm3.72}$ | $90.56_{\pm1.99}$ | $85.51_{\pm1.02}$ | $72.09_{\pm1.77}$ | $64.74_{\pm1.44}$ | $84.02_{\pm0.48}$ | $76.37_{\pm2.22}$ |
| Impr. | **+4.48** | **+1.83** | **+1.21** | **+0.61** | **+0.66** | **+1.12** | **+1.07** | **+0.53** |

Table 2: Performance comparison on molecular property prediction classification tasks. All scores are represented as "mean$_{\pm\text{std}}$" with three different seeds. The third lines (Impr.) denote performance improvement with the *APMP* module. Note that certain tasks for N-GRAM are not reported (denoted by "-") due to the excessive time required by its original vertex embedding training strategy. Similarly, some tasks for MolCLR are excluded due to severe validation/test imbalance, making evaluations infeasible.

## Quantitative Performance Comparison

For the downstream molecular property prediction tasks, Table 1 and Table 2 present the overall performance results on 3 regression tasks and 8 classification tasks across 8 advanced MFMs. The performance comparisons provide the following observations: (1) All MFMs with the proposed plugin consistently outperform the baselines on all datasets with a large margin on most of them. The relative improvement across all datasets and MFMs reaches up to 4.48% for classification tasks, while for regression tasks, it is 19.20%. This remarkable boosting validates the effectiveness and generalizability of the integrated plugin to enhance the original pretrained representation of atoms. (2) For the regression tasks, the improvements through the proposed method are particularly pronounced. Especially in the small dataset FreeSolv, which contains only 642 labeled molecules, the plugin achieves a 19.20% relative RMSE improvement over the baseline MFM. This confirms the efficacy of the proposed method, as it can significantly enhance performance even when label information is scarce. (3) For the classification tasks, the improvement on single-task datasets gen-

erally surpasses that on multi-task datasets. This is due to the fact that each task may have its specific patterns, while in a multi-task framework, the confusion caused by common patterns can interfere with individual task performance, even though the overall performance still exceeds that of the baseline MFMs.

## Interpretability of BBBP Pattern Library

Based on the CPC calculation, we construct the property-specific pattern library for the blood-brain barrier (BBB) penetration. As illustrated in Figure 3, starting from certain atom node (red star) in the Zafuleptine, we obtain two groups of sampling paths with their $\delta$ scores. For the left column, the connection of the phenyl group and fluorine group denotes the common pattern (orange regions), and for the right column the significant impact of the amine group (yellow regions) is highlighted for the atom representation. In contrast, the last row shows the paths with smaller $\delta$ values (gray regions), indicating minimal contribution to the atom representation. The above observations are intuitive: (1) The conjugated system and non-polar nature of the ben-
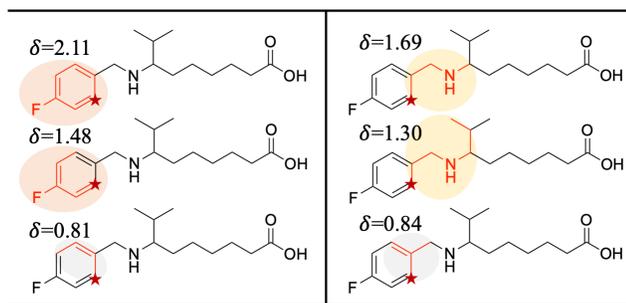
Figure 3: Interpretability cases of the BBBP pattern library. The two columns denote the common patterns and negligible paths starting from certain atom $v$ after two rounds of path sampling, respectively. Here, we use $\delta = \text{CPC}(v) * n$ to represent the relative contribution to the node representation.

zene ring endow Zafuleptine with high lipophilicity (Abbott et al. 2010; Geldenhuys et al. 2015; Cornelissen et al. 2023), enhancing its interaction with lipid components of cell membranes and promoting passive diffusion through lipophilic and hydrophilic phases across the BBB. (2) The fluorine group introduces hydrophobic properties, further increasing the molecule's lipophilicity and its ability for passive membrane diffusion (Sun and Adejare 2006; Muller, Faeh, and Diederich 2007). Notably, the robust C-F bond resists metabolic degradation, reducing the likelihood of the molecule being broken down before crossing the BBB. (3) In the context of active transport across the BBB, the amine groups can act as hydrogen bond receptors (Young et al. 1988; Pardridge 2012; Geldenhuys et al. 2015), which is crucial for binding to influx or efflux transporters involved in the BBB transport. Additional interpretability cases and the corresponding discussions can be found at Appendix C.3.

## Ablation Studies

To validate the impact of hyperparameters on the APMP module for **RQ1**, we conduct the ablation studies on two key parameters, namely the filtering ratio of path candidates $\beta = n/N$ and the length of sampled patterns $L$. Further, to investigate the mechanisms by which the plugin functions for **RQ2**, we design the ablation studies regarding where the plugin is integrated into the MFMs and how it is trained.

**Filtering Ratio of Path Candidates** $\beta$. Figure 4a shows the prediction performance varying with top $\beta\%$ confidence of ranked sampled paths as high-confidence patterns. The performance reaches its highest and most stable levels when approximately top 30% to 45% of the paths are selected. In contrast, the performance declines when fewer than 20% or more than 80% of paths are chosen, although it still exceeds baseline without APMP. This finding underscores the need for a balanced path selection to avoid information loss or redundancy, highlighting the effectiveness of the CPF block.

**Length of Sampled Path** $L$. Figure 4b shows prediction performance varying with different lengths of sampling paths. When $L$ is relatively larger ($L \geq 6$), the prediction performance surpasses compared to these shorter lengths
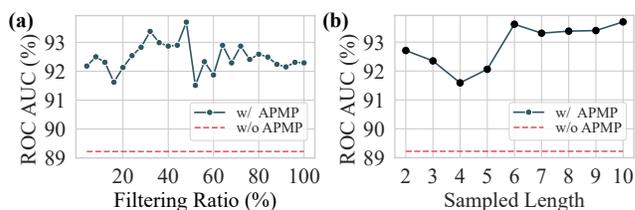


Figure 4: Ablation results on the hyperparameters of APMP for the BBBP property task, including: (a) filtering ratio of path candidates $\beta$, and (b) length of sampled patterns $L$.

| APMP Feature Integration | | Finetune | | ROC-AUC |
|---|---|---|---|---|
| Before encoder | After encoder | Yes | No | |
| ✓ | | | ✓ | $89.22_{\pm 1.58}$ |
| | | | ✓ | $74.09_{\pm 0.89}$ |
| | ✓ | | ✓ | $89.24_{\pm 1.57}$ |
| ✓ | | ✓ | | $88.19_{\pm 3.00}$ |
| | ✓ | ✓ | | $\mathbf{93.70_{\pm 0.23}}$ |

Table 3: Ablation results on where to integrate the APMP plugin into the baseline MFM Unimol and how it is trained for the downstream BBBP property classification task.

($L<6$), showing that sufficiently long paths can encompass a more diverse and expressive range of information. This finding can facilitate the discovery of common patterns and subsequently enhance the overall performance.

**Plugin Integration Workflow.** Table 3 demonstrates that the best performance is yielded by finetuning combined representations from MFM encoders and the APMP plugin. Specifically, the pretrained MFM encoders generate the atom representations, which are enhanced by the APMP that generates the representations from common patterns, integrates them with the outputs of MFM encoders, and finetunes the combined representations via a predictor. In contrast, introducing plugin-generated features before input would require substantial additional training for the MFM encoders, potentially necessitating training from scratch, complicating rapid convergence. Additionally, if the plugin is added directly without finetuning, the representations will be quite random and difficult to improve performance.

## Conclusion

In this paper, we introduce a novel association pattern-aware message passing method to uncover the molecular property-specific common patterns and expand the ability to represent complex interactions within molecular graphs. The method functions as a general plugin that can be integrated into the MFMs with pretrained encoders, effectively enhancing atom representations and thereby increasing their expressiveness. Extensive tests on 11 benchmark molecular property tasks across 8 advanced MFMs show the effectiveness of the proposed plugin. Furthermore, the construction of property-specific pattern library highlights the strong interpretability for further potential application in the real-world scenarios.

17605

## Acknowledgments

## References

Abbott, N. J.; Patabendige, A. A.; Dolman, D. E.; Yusof, S. R.; and Begley, D. J. 2010. Structure and function of the blood–brain barrier. *Neurobiology of disease*, 37(1): 13–25.

Al-Hasani, R.; and Bruchas, M. R. 2011. Molecular mechanisms of opioid receptor-dependent signaling and behavior. *The Journal of the American Society of Anesthesiologists*, 115(6): 1363–1381.

Brin, S.; and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7): 107–117.

Cornelissen, F. M.; Markert, G.; Deutsch, G.; Antonara, M.; Faaij, N.; Bartelink, I.; Noske, D.; Vandertop, W. P.; Bender, A.; and Westerman, B. A. 2023. Explaining blood–brain barrier permeability of small molecules by integrated analysis of different transport mechanisms. *Journal of Medicinal Chemistry*, 66(11): 7253–7267.

DiMasi, J. A.; Grabowski, H. G.; and Hansen, R. W. 2016. Innovation in the pharmaceutical industry: new estimates of R&D costs. *Journal of health economics*, 47: 20–33.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining(KDD)*, 135–144.

Fang, X.; Liu, L.; Lei, J.; He, D.; Zhang, S.; Zhou, J.; Wang, F.; Wu, H.; and Wang, H. 2022. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2): 127–134.

Geldenhuys, W. J.; Mohammad, A. S.; Adkins, C. E.; and Lockman, P. R. 2015. Molecular determinants of blood–brain barrier permeation. *Therapeutic delivery*, 6(8): 961–971.

Glaser, R. 2012. *Biophysics: an introduction*. Springer Science & Business Media.

Jeh, G.; and Widom, J. 2002. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining(KDD)*, 538–543.

Jeh, G.; and Widom, J. 2003. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web(WWW)*, 271–279.

Jia, L.; Feng, Z.; Zhang, H.; Song, J.; Zhong, Z.; Yao, S.; and Song, M. 2022. Explainable Fragment-Based Molecular Property Attribution. *Advanced Intelligent Systems*, 4(10): 2200104.

Jia, L.; Ying, Y.; Feng, Z.; Zhong, Z.; Yao, S.; Hu, J.; Duan, M.; Wang, X.; Song, J.; and Song, M. 2024. Association

Pattern-aware Fusion for Biological Entity Relationship Prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Kalgutkar, A. S. 2019. Designing around structural alerts in drug discovery. *Journal of Medicinal Chemistry*, 63(12): 6276–6302.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations(ICLR)*.

Liu, S.; Demirel, M. F.; and Liang, Y. 2019. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32.

Liu, S.; Wang, H.; Liu, W.; Lasenby, J.; Guo, H.; and Tang, J. 2022. Pre-training Molecular Graph Representation with 3D Geometry. In *International Conference on Learning Representations*.

Muller, K.; Faeh, C.; and Diederich, F. 2007. Fluorine in pharmaceuticals: looking beyond intuition. *science*, 317(5846): 1881–1886.

Nelson, S. D. 1982. Metabolic activation and drug toxicity. *Journal of medicinal chemistry*, 25(7): 753–765.

Pardridge, W. M. 2012. Drug transport across the blood–brain barrier. *Journal of cerebral blood flow & metabolism*, 32(11): 1959–1972.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining(KDD)*, 701–710.

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. *Advances in neural information processing systems*, 33: 12559–12571.

Silbey, R. J.; Alberty, R. A.; Papadantonakis, G. A.; and Bawendi, M. G. 2022. *Physical chemistry*. John Wiley & Sons.

Smith, K. S.; Smith, P. L.; Heady, T. N.; Trugman, J. M.; Harman, W. D.; and Macdonald, T. L. 2003. In vitro metabolism of tolcapone to reactive intermediates: relevance to tolcapone liver toxicity. *Chemical research in toxicology*, 16(2): 123–128.

Sun, S.; and Adejare, A. 2006. Fluorinated molecules as drugs and imaging agents in the CNS. *Current topics in medicinal chemistry*, 6(14): 1457–1464.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11): 992–1003.

Tu, K.; Cui, P.; Wang, X.; Wang, F.; and Zhu, W. 2018. Structural deep embedding for hyper-networks. In *Proceedings of the AAAI conference on artificial intelligence(AAAI)*, 426–433.

Usui, T.; Mise, M.; Hashizume, T.; Yabuki, M.; and Komuro, S. 2009. Evaluation of the potential for drug-induced liver injury based on in vitro covalent binding to human liver

proteins. *Drug Metabolism and Disposition*, 37(12): 2383–2392.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems(NeurIPS)*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations(ICLR)*.

Wang, Y.; Wang, J.; Cao, Z.; and Barati Farimani, A. 2022. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3): 279–287.

Wang, Y.; Zhang, J.; Huang, Z.; Li, W.; Feng, S.; Ma, Z.; Sun, Y.; Yu, D.; Dong, F.; Jin, J.; et al. 2023. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023(WWW)*, 406–416.

Widmaier, E.; Raff, H.; and Strang, K. T. 2022. *Vander's human physiology*. McGraw-Hill US Higher Ed USE.

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.

Xia, F.; Liu, J.; Nie, H.; Fu, Y.; Wan, L.; and Kong, X. 2019. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(2): 95–107.

Xia, J.; Zhu, Y.; Du, Y.; and Li, S. Z. 2023. A systematic survey of chemical pre-trained models. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6787–6795.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How powerful are graph neural networks? In *International Conference on Learning Representations(ICLR)*.

Yang, Z.; Zhong, W.; Lv, Q.; and Chen, C. Y.-C. 2022. Learning size-adaptive molecular substructures for explainable drug–drug interaction prediction by substructure-aware graph neural network. *Chemical science*, 13(29): 8693–8703.

Yao, S.; Feng, Z.; Song, J.; Jia, L.; Zhong, Z.; and Song, M. 2022. Chemical property relation guided few-shot molecular property prediction. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Young, R. C.; Mitchell, R. C.; Brown, T. H.; Ganellin, C. R.; Griffiths, R.; Jones, M.; Rana, K. K.; Saunders, D.; and Smith, I. R. 1988. Development of a new physicochemical model for brain penetration and its application to the design of centrally acting H2 receptor histamine antagonists. *Journal of medicinal chemistry*, 31(3): 656–671.

Zhang, R.; Zou, Y.; and Ma, J. 2020. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations (ICLR)*.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C.-K. 2021. Motif-based graph self-supervised learning for molecular

property prediction. *Advances in Neural Information Processing Systems*, 34: 15870–15882.

Zhou, G.; Gao, Z.; Ding, Q.; Zheng, H.; Xu, H.; Wei, Z.; Zhang, L.; and Ke, G. 2023. Uni-Mol: A Universal 3D Molecular Representation Learning Framework. In *The Eleventh International Conference on Learning Representations*.