Motif-aware Attribute Masking for Molecular Graph Pre-training

Eric Inae
University of Notre Dame
einae@nd.edu

Gang Liu
University of Notre Dame
gliu7@nd.edu

Meng Jiang
University of Notre Dame
mjiang2@nd.edu

Abstract

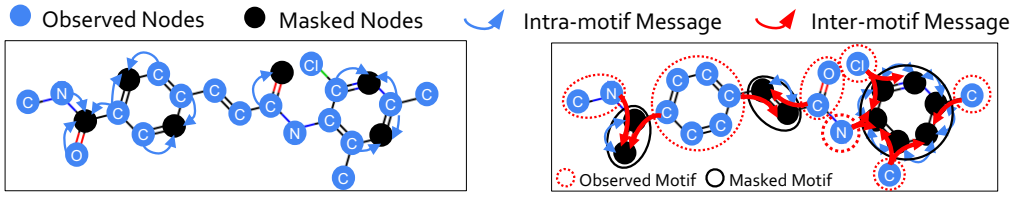
Attribute reconstruction is used to predict node or edge features in the pre-training of graph neural networks. Given a large number of molecules, they learn to capture structural knowledge, which is transferable for various downstream property prediction tasks and vital in chemistry, biomedicine, and material science. Previous strategies that randomly select nodes to do attribute masking leverage the information of local neighbors. However, the over-reliance of these neighbors inhibits the model's ability to learn long-range dependencies from higher-level substructures, such as functional groups or chemical motifs. To explicitly measure and encourage the inter-motif knowledge transfer in pre-trained models, we define inter-motif node influence measures and propose a novel motif-aware attribute masking strategy to capture long-range inter-motif structures by leveraging the information of atoms in neighboring motifs. Once each graph is decomposed into disjoint motifs, the features for every node within a sample motif are masked and subsequently predicted using a graph decoder. We evaluate our approach on eleven molecular classification and regression datasets and demonstrate its advantages.

1 Introduction

Molecular property prediction has been an important topic of study in fields such as physical chemistry, physiology, and biophysics [1]. It can be defined as a graph label prediction problem and addressed by machine learning. However, graph learning models such as graph neural networks (GNNs) must overcome issues in data scarcity, as the creation and testing of real-world molecules is an expensive endeavor [2]. To address labeled data scarcity, model pre-training has been utilized as a fruitful strategy for improving a model's predictive performance on downstream tasks, as pre-training allows for the transfer of knowledge from large amounts of unlabeled data. The selection of pre-training strategy is still an open question, with contrastive tasks [3] and predictive/generative tasks [4] being the most popular methods.

Attribute reconstruction is one predictive method for graphs that utilizes masked autoencoders to predict node or edge features [4, 6, 7]. Masked autoencoders have found success in vision and language domains [8, 9] and have been adopted as a pre-training objective for graphs as the reconstruction task is able to transfer structural pattern knowledge [4], which is vital for learning specific domain knowledge such as valency in material science. Additional domain knowledge which is important for molecular property prediction is that of functional groups, also called chemical motifs [10]. *The presence and interactions between chemical motifs directly influence molecular properties, such as reactivity and solubility* [11, 12]. Prior work in message passing for quantum chemistry has shown that long-range dependencies are important for downstream prediction in chemical domains [13]. Additional works show that long-range interactions are vital in small molecule property prediction [14–16]. So, to enable the transfer of both long-range information and motif interactions, it would be important to transfer inter-motif knowledge during the pre-training of graph neural networks.

Inae et al., Motif-aware Attribute Masking for Molecular Graph Pre-training. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR XXX, Virtual Event, November 26–29, 2024.



(a) Random [4, 5] **(b)** Ours, named MoAMa

Figure 1: Our MoAMa masks every node in sampled motifs to pre-train GNNs. The full masking of a motif forces the GNNs to learn to (1) pass feature information across motifs and (2) pass local structural information within motifs. Compared to the traditional random attribute masking, the motif-aware masking captures more essential domain knowledge to learn graph embeddings. Random masking is unaware of chemical knowledge, leading to poor transferability in molecular tasks.

Unfortunately, the random attribute masking strategies used in previous work for graph pre-training were not able to capture the long-range dependencies inherent in inter-motif knowledge [6, 17, 18]. That is because they rely on neighboring node feature information for reconstruction [4, 5]. Notably, leveraging the features of local neighbors can contribute to learning important local information, including valency and atomic bonding. However, GNNs heavily rely on the neighboring node's features rather than graph structure [19], and this over-reliance inhibits the model's ability to learn from motif structures as message aggregation will prioritize local node feature information due to the propagation bottleneck [20, 21]. For example, as shown on the left-hand side of Figure 1, if only a (small) partial set of nodes were masked in several motifs, the pre-trained GNNs would learn to predict the node types (i.e., carbon) of two atoms in the benzene ring based on the features and structure of the other four carbon atoms in the ring, limiting the knowledge transfer of long-range dependencies. To measure the inter-motif knowledge transfer of graph pre-training strategies, we define five inter-motif influence measurements and report our findings in Section 6: Basically, the inter-motif influence is strongly correlated with property prediction accuracy.

Recent successes in vision and language domains have shown the utility of masking semantically related regions, such as pixel batches [22–24] and multi-token spans [25–27], and have demonstrated that a random masking strategy is not guaranteed to transfer necessary inter-part relations and intra-part patterns [23]. To better enable the transfer of long-range inter-part relations downstream, we propose a novel semantically-guided masking strategy based on chemical motifs. In Figure 1, we visually demonstrate our method for motif-aware attribute masking, where each molecular graph is decomposed into disjoint motifs. Then the node features for each node within the motif will be masked by a mask token. A graph decoder will predict the masked features of each node within the motif as the reconstruction task. The benefits of this strategy are twofold. First, because all features of the nodes within the motif are masked, our strategy reduces the amount of feature information being passed within the motif and relieves the propagation bottleneck, allowing for the greater transfer of inter-motif feature and structural information. Second, the masking of all intra-motif node features explicitly forces the decoder to transfer intra-motif structural information. A novel graph pre-training solution based on the **Mo**tif-aware **A**ttribute **Ma**sking strategy, called **MoAMa**, is able to learn long-range inter-motif dependencies with knowledge of intra-motif structure. We evaluate our strategy on eight molecular property prediction datasets and demonstrate its improvement to inter-motif knowledge transfer as compared to previous strategies.

2 Related Work

Molecular graph pre-training. The prediction of molecular properties based on graphs is important [1]. Molecules are scientific data that are time- and computation-intensive to collect and annotate for different property prediction tasks [28]. Many self-supervised learning methods [4, 5, 29–31] were proposed to capture the transferable knowledge from another large scale of molecules without annotations. For example, AttrMask [4] randomly masked atom attributes for prediction. GraphMAE [5] pre-trained the prediction model with generative tasks to reconstruct node and edge attributes. D-SLA [30] used contrastive learning based on graph edit distance. These pre-training tasks could not well capture useful knowledge for various domain-specific tasks since they fail to

incorporate important domain knowledge in pre-training. A great line of prior work [29, 32, 33] used graph motifs which are the recurrent and statistically significant subgraphs to characterize the domain knowledge contained in molecular graph structures, e.g., functional groups. However, their solutions were tailored to specific frameworks for either generation-based or contrast-based molecular pre-training. Additionally, explicit motif type generation/prediction inherently does not transfer intra-motif structural information and is computationally expensive due to the large number of prediction classes. In this work, we study on the strategies of attribute masking with the awareness of domain knowledge (i.e., motifs), which plays an essential role in self-supervised learning frameworks [31].

Masking strategies on molecules. Attribute masking of atom nodes is a popular method in graph pre-training given its broad usage in predictive, generative, and contrastive self-supervised tasks [4, 5, 18, 34, 35]. For example, predictive and generative pre-training tasks [4, 5, 31] mask atom attributes for prediction and reconstruction. Contrastive pre-training tasks [34, 35] mask nodes to create another data view for alignment. Despite the widespread use of attribute masking in molecular pre-training, there is a notable absence of comprehensive research on its strategy and effectiveness. Previous studies have largely adopted strategies from the vision and language domains [8, 9], where atom attributes are randomly masked with a predetermined ratio. Since molecules are atoms held together by strict chemical rules, the data modality of molecular graphs is essentially different from natural images and languages. For molecular graphs, random attribute masking results in either over-reliance on intra-motif neighbors [36] or breaking the inter-motif connections via random edge masking. In this work, we introduce a novel strategy of attribute masking, which turns out to capture and transfer useful knowledge from intra-motif structures and long-range inter-motif node features.

3 Preliminaries

Graph property prediction. Given a graph $G = (\mathcal{V}, \mathcal{E}) \in \mathcal{G}$ with the node set $\mathcal{V}$ for atoms and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ for bonds, we have a d-dimensional node attribute matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ that represents atom features such as atom type and chirality. We use $y \in \mathcal{Y}$ as the graph-level property label for G, where $\mathcal{Y}$ represents the label space. A predictor with the encoder-decoder architecture is trained to encode G into a representation vector in the latent space and decode the representation to predict $\hat{y}$. The training process optimizes the parameters to make $\hat{y}$ to be the same as the true label value y. A GNN is a commonly used encoder that generates k-dimensional node representation vectors, denoted as $\mathbf{h}_v \in \mathbb{R}^k$, for any node $v \in \mathcal{V}$:

$$\mathbf{H} = \{\mathbf{h}_v : v \in \mathcal{V}\} = \text{GNN}(G) \in \mathbb{R}^{|\mathcal{V}| \times k}. \tag{1}$$

Here $\mathbf{H}$ is the node representation matrix for the graph G. Without loss of generality, we implement Graph Isomorphism Networks (GIN) [37] as the choice of GNN in accordance with previous work [4]. Once the set of node representations are created, a READOUT$(\cdot)$ function (such as max, mean, or sum) is used to summarize the node-level representation into graph-level representation $\mathbf{h}_G$ for any G:

$$\mathbf{h}_G = \text{READOUT}(\mathbf{H}) \in \mathbb{R}^k. \tag{2}$$

The graph-level representation vector $\mathbf{h}_G$ is subsequently passed through a multi-layer perceptron (MLP) to generate the label prediction $\hat{y}$, which exists in the label space $\mathcal{Y}$:

$$\hat{y} = \text{MLP}(\mathbf{h}_G) \in \mathcal{Y}. \tag{3}$$

GNN pre-training. Random initialization of the predictor's parameters would easily result in suboptimal solutions for graph property prediction. This is because the number of labeled graphs is usually small. It prevents a proper coverage of task-specific graph and label spaces [4, 28]. To improve generalization, GNN pre-training is often used to warm-up the model parameters based on a much larger set of molecules without labels. In this work, we focus on the attribute masking strategy for GNN pre-training that aims to predict the masked values of node attributes given the unlabeled graphs.

Molecular motifs. Each graph G can be decomposed into disjoint subgraphs, or motifs. We denote the decomposition result as $\mathcal{M}_G = \{M_1, M_2, ..., M_n\}$, which is a set of n motifs. Each motif $M_i = (\mathcal{V}_i, \mathcal{E}_i)$, for $i \in \{1, 2, ..., n\}$, is a disjoint subgraph of G such that $\mathcal{V}_i \subset \mathcal{V}$ and $\mathcal{E}_i \subset \mathcal{E}$. For each motif multi-set $\mathcal{M}_G$, the union of all motifs $M_i \in \mathcal{M}_G$ should equal G. Formally, this means

$\mathcal{V} = \bigcup_i V_i$ and $\mathcal{E} = (\bigcup_i E_i) \bigcup E_x$, where E_x represents all the edges removed between motifs during decomposition.

We leverage chemical domain knowledge by using the BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) algorithm [38]. This algorithm follows 16 rules for decomposition, which define the bonds that should be cleaved from the molecule in order to create the multi-set of motifs. In practice, these motifs may be quite large, so we perform additional decomposition as described in previous work by isolating rings and further reducing motifs with a node of degree greater than two [29]. Two key strengths of the BRICS algorithm over a motif-mining strategy [39] is that no training is required and important structural features, such as rings, are inherently preserved.

4 Inter-Motif Influence

To measure the influence generally from (either intra-motif or inter-motif) source nodes on a target node v, we design a measure that quantifies the influence from any source node u in the same graph G, denoted by $s(u, v)$. $\mathbf{h}_v$ was learned by Eq. (1) and was influenced by node u. When the embedding of u is eliminated from GNN initialization, i.e., set $\mathbf{h}_u^{(0)} = \vec{0}$, Eq. (1) would produce a new representation vector of node v, denoted by $\mathbf{h}_{v,\text{w/o } u}$. We define the influence using L^2-norm:

$$s(u, v) = \|\mathbf{h}_v - \mathbf{h}_{v,\text{w/o } u}\|_2. \tag{4}$$

The collective influence from a group of nodes in a motif $M = (\mathcal{V}_M, \mathcal{E}_M)$ is measured as follows:

$$s_{\text{motif}}(v, M) = \frac{1}{|\mathcal{V}_M \setminus \{v\}|} \sum_{u \in \mathcal{V}_M \setminus \{v\}} s(u, v). \tag{5}$$

Suppose the target node v is in the motif $M_v = (\mathcal{V}_{M_v}, \mathcal{E}_{M_v})$. Using M_v as the target motif, the influence from intra-motif and inter-motif nodes can be calculated as:

$$s_{\text{intra}}(v) = s_{\text{motif}}(v, M_v);\ s_{\text{inter}}(v) = \frac{\sum_{M \in \mathcal{M} \setminus \{M_v\}} |\mathcal{V}_M| \times s_{\text{motif}}(v, M)}{|\mathcal{V} \setminus \mathcal{V}_{M_v}|}. \tag{6}$$

Usually the number of inter-motif nodes is significantly bigger than the number of intra-motif nodes, i.e., $|\mathcal{V}| \gg |\mathcal{V}_{M_v}|$, which reveals two issues in the influence measurements. First, when the target motif is too small (e.g., has only one or two nodes), the intra-motif influence cannot be defined or is defined on the interaction with only one neighbor node. Second, most inter-motif nodes are not expected to have any influence, so the average function in Eq. (5) would lead comparisons to be biased to intra-motif influence. To address the two issues, we constrain the influence summation to be on the *same number* of nodes (i.e., top-k) from the intra-motif and inter-motif node groups. Explicitly, this means $u \in \mathcal{V}_M / \{v\}$ in Eq. (5) is sampled from the top-k most influencial nodes (top-3). The ratio of inter- to intra-motif influence over the graph dataset $\mathcal{G}$ is defined as:

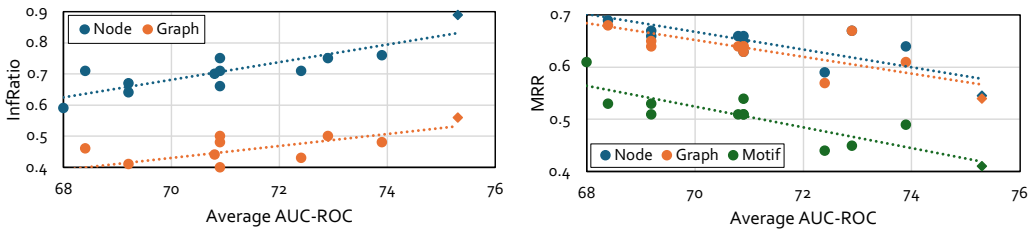
$$\text{InfRatio}_{\text{node}} = \frac{1}{\sum_{(\mathcal{V},\mathcal{E}) \in \mathcal{G}} |\mathcal{V}|} \sum_{(\mathcal{V},\mathcal{E}) \in \mathcal{G}} \sum_{v \in \mathcal{V}} \frac{s_{\text{inter}}(v)}{s_{\text{intra}}(v)} \tag{7}$$

$$\text{InfRatio}_{\text{graph}} = \frac{1}{|\mathcal{G}|} \sum_{G=(\mathcal{V},\mathcal{E}) \in \mathcal{G}} \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{s_{\text{inter}}(v)}{s_{\text{intra}}(v)} \tag{8}$$

where the average function is performed at the node level and graph level, respectively. Eq. (7) directly measures the influence ratios of all nodes v within the dataset $\mathcal{G}$. However, this measure may include bias due to the distribution of nodes within each graph. We alleviate this bias in Eq. (8) by averaging influence ratios across each graph first.

While the InfRatio measurements compare general inter- and intra-motif influences, these measures combine all inter-motif nodes into one set and do not consider the number of motifs in each graph. We define rank-based measures that consider the distribution of motif counts across $\mathcal{G}$.

Let $\{M_1, ..., M_i, ..., M_n\}$ be an ordered set, where $M_i \in \mathcal{M}$ and $s_{\text{motif}}(v, M_i) \geq s_{\text{motif}}(v, M_j)$ if $i < j$. If $M_i = M_v$, we define $\text{rank}_v = i$. Note that graphs with only one motif are excluded as the distinction



(a) InfRatio vs. Model Accuracy **(b)** MRR vs. Model Accuracy

Figure 2: InfRatio and MRR measurements of pre-trained models against average AUC-ROC scores. Each point represents a different pre-trained model, with ◆ denoting MoAMa. The InfRatio and MRR measurements demonstrate strong positive/negative correlations. These trends reveal performance can be correlated with inter-motif knowledge transfer for attribute masking strategies.

between inter and intra-motif nodes loses meaning. From this ranking, we define our score for inter-motif node influence averaged at the node, motif, and graph levels, derived from a similar score measurement used in information retrieval, Mean Reciprocal Rank (MRR) [40]. Similar to the InfRatio measurements, MRR_{node} directly captures the impact of the influence ranks for each node within the full graph set, whereas $\text{MRR}_{\text{graph}}$ alleviates bias on the number of nodes within a graph by averaging across individual graphs first. Because these rank-based measurements are intrinsically dependent on the number of motifs within each graph, we additionally define $\text{MRR}_{\text{motif}}$ which weights the measurement towards popular motif counts within the data distribution.

$$\text{MRR}_{\text{node}} = \frac{1}{\sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}} |\mathcal{V}|} \sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}} \sum_{v\in\mathcal{V}} \frac{1}{\text{rank}_v} \tag{9}$$

$$\text{MRR}_{\text{graph}} = \frac{1}{|\mathcal{G}|} \sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}} \frac{1}{|\mathcal{V}|} \sum_{v\in\mathcal{V}} \frac{1}{\text{rank}_v} \tag{10}$$

$$\text{MRR}_{\text{motif}} = \sum_{n=2}^{N} \frac{|\mathcal{G}^{(n)}|}{|\mathcal{G}| \sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}^{(n)}} |\mathcal{V}|} \sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}^{(n)}} \sum_{v\in\mathcal{V}} \frac{1}{\text{rank}_v}, \tag{11}$$

where $\mathcal{G}^{(n)}$ is the set of graphs that contain $n \in [2, N]$ motifs.

In information retrieval, MRR scores are used to quantify how well a system can return the most relevant item for a given query. Higher MRR scores indicate that relevant items were returned at higher ranks for each query. However, as opposed to traditional MRR measurements, where a higher rank for the most relevant item indicates better performance, lower scores are preferred for our MRR measurements as lower intra-motif influence rank indicate greater inter-motif node influence. In Fig. 2, we present the InfRatio and MRR measurements of different pre-trained attribute reconstruction models against their average AUC-ROC over different classification tasks. Details for baseline models and tasks can be found in Sec A.1. For the InfRatio measurements, we see strong positive correlations with Pearson coefficients of 0.85 and 0.76, and for the MRR measurements, we see strong negative correlations with coefficients of -0.80, -0.78, and -0.85. These strong correlations suggest that performance increases given higher inter-motif knowledge transfer in attribute reconstruction-based methods. This result motivates our work as MoAMa was specifically designed to increase inter-motif knowledge transfer and demonstrates the greatest transfer amongst the baselines.

5 Proposed Solution

In this section, we present our novel solution named MoAMa for pre-training graph neural networks on molecular data. We will give details about the strategy of motif-aware attribute masking and reconstruction. Each molecule G will have some portion of their nodes masked according to domain knowledge based motifs. We replace the attributes of all masked nodes with a special mask token. Then, the GNN in Eq. (1) encodes the masked graph to the node representation space, and an MLP reconstructs the atom types for each masked atom. Algorithm 1 presents details of full framework.

Algorithm 1 MoAMa Pre-training Method

Input: Graph set $\mathcal{G}$, encoder $f_\theta(\cdot)$, attribute decoder $g_\phi(\cdot)$, motif decomposition BRICS$(\cdot)$, masking strategy MASK$(\cdot)$, hyperparameter β
Output: Learned encoder $f_\theta(\cdot)$
for $G_i, G_j \in \mathcal{G}$ s.t. $G_i \neq G_j$ **do**
  $\mathcal{M}_{G_i} \leftarrow \text{BRICS}(G_i)$
  Sample $\mathcal{M}'_{G_i} \sim \mathcal{M}_{G_i}$
  $G_{i_{[\text{MASK}]}} \leftarrow \text{MASK}(\mathcal{M}'_{G_i})$
  $\mathbf{H} \leftarrow f(G_{i_{[\text{MASK}]}})$
  Calculate $\mathcal{L}_{\text{rec}}$ using $g_\phi(\cdot)$ according to Eq. (13)
  Calculate $\mathcal{L}_{\text{aux}}$ according to Eq. (14)
  Calculate $\mathcal{L} \leftarrow \beta\mathcal{L}_{\text{rec}} + (1-\beta)\mathcal{L}_{\text{aux}}$
  Update θ and ϕ
end for
return $f_\theta(\cdot)$

5.1 Motif-aware Attribute Masking and Reconstruction

To perform motif-aware attribute masking, m motifs are sampled to form the multi-set $\mathcal{M}'_G \subset \mathcal{M}_G$ such that $(\sum_{(\mathcal{V}_i,\mathcal{E}_i)\in\mathcal{M}'_G} |\mathcal{V}_i|)/|\mathcal{V}| = \alpha$, for α is a chosen ratio value. In order to guarantee the availability of inter-motif node information, we assert that sampled motifs may not be adjacent. Additionally, if motifs exceed a certain size ($|\mathcal{V}| > 9$), then it is possible that several nodes will not receive any meaningful feature information during message passing. To alleviate the difficulty of the reconstruction task in this specific case, we reintroduce node features to one or more nodes in the masked motif to guarantee meaningful feature knowledge is passed to all nodes. This case is rare however, only accounting for 0.7% of motifs in the dataset. We choose $\alpha = 0.25$ as it follows the settings of previous work [5].

Given a selected motif $M \in \mathcal{M}'_G$, nodes within M have their attributes masked with a mask token [MASK], which is a vector $\mathbf{m} \in \mathbb{R}^d$. Each element in $\mathbf{m}$ is a special value that is not present within the attribute space for that particular dimension. We use $\mathcal{V}_{[\text{MASK}]} = \{v \in \mathcal{V}_i : M_i = (\mathcal{V}_i, \mathcal{E}_i) \in \mathcal{M}'_G\}$ to denote the set of all the masked nodes. We then define the input node features in the masked attribute matrix $\mathbf{X}_{[\text{MASK}]} \in \mathbb{R}^{|\mathcal{V}|\times d}$ for any $v \in \mathcal{V}$ using the following equation:

$$(\mathbf{X}_{[\text{MASK}]})_v = \begin{cases} \mathbf{X}_v, & v \notin \mathcal{V}_{[\text{MASK}]}, \\ \mathbf{m}, & v \in \mathcal{V}_{[\text{MASK}]}, \end{cases} \tag{12}$$

where $(\mathbf{X}_{[\text{MASK}]})_v$ and $\mathbf{X}_v$ denote the row of the node v in $\mathbf{X}_{[\text{MASK}]}$ and $\mathbf{X}$, respectively. With a GNN encoder, all nodes with attributes $\mathbf{X}_{[\text{MASK}]}$ for the masked graph $G_{[\text{MASK}]}$ are encoded to the latent representation space according to Eq. (1): $\mathbf{H} = \text{GNN}(G_{[\text{MASK}]})$. $\mathbf{H}$ is then used to define the reconstruction loss of the node attributes:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{v\in\mathcal{V}_{[\text{MASK}]}}[\log p(\mathbf{X}|\mathbf{H})], \tag{13}$$

where $p(\mathbf{X}|\mathbf{H})$ for the reconstruction attribute value is inferred by a decoder. In practice, reconstruction loss is measured using the scaled cosine error (SCE) [5], which calculates the difference between the probability distribution for the reconstruction attributes and the one-hot encoded target label vector. This choice of reconstruction loss is further discussed in Appendix A.2.

Because attribute masking focuses on local graph structures and suffers from representation collapse [4, 5], we use a knowledge-enhanced auxiliary loss $\mathcal{L}_{\text{aux}}$ to complement $\mathcal{L}_{\text{rec}}$. Given any two graphs G_i and G_j from the graph-based chemical space $\mathcal{G}$, $\mathcal{L}_{\text{aux}}$ first calculates the Tanimoto similarity [41] between G_i and G_j as $\text{T}(G_i, G_j)$ based on the bit-wise fingerprints, which characterizes frequent fragments in the molecular graphs. Then $\mathcal{L}_{\text{aux}}$ aligns the latent representations with the Tanimoto similarity using the cosine similarity, inspired by previous work [42]. Formally, we define:

$$\mathcal{L}_{\text{aux}} = \sum_{1\leq i,j\leq|\mathcal{G}|, i\neq j,} \Big(\text{T}(G_i, G_j) - \cos(\mathbf{h}_{G_i}, \mathbf{h}_{G_j})\Big)^2 \tag{14}$$

Table 1: Test AUC (%) performance on eight Open Graph Benchmark (OGB) molecular classification datasets. The best AUC-ROC values for each dataset are in **bold**. Models with (*) use the same GNN architecture as MoAMa.

	MUV	ClinTox	SIDER	HIV	Tox21	BACE	ToxCast	BBBP	**Avg**
No Pretrain*	70.7±1.8	58.4±6.4	58.2±1.7	75.5±0.8	74.6±0.4	72.4±3.8	61.7±0.5	65.7±3.3	67.2
MGSSL* [29]	77.6±0.4	77.1±4.5	61.6±1.0	75.8±0.4	75.2±0.6	78.8±0.9	63.3±0.5	68.8±0.9	72.3
MCM [43]	74.4±0.6	64.7±0.5	62.3±0.9	72.7±0.3	74.4±0.1	79.5±1.3	61.0±0.4	71.6±0.6	69.7
Grover [32]	50.6±0.4	75.4±8.6	57.1±1.6	67.1±0.3	76.3±0.6	79.5±1.1	63.4±0.6	68.0±1.5	67.2
AttrMask* [4]	75.8 ±1.0	73.5±4.3	60.5±0.9	75.3±1.5	75.1±0.9	77.8±1.8	63.3±0.6	65.2±1.4	70.8
ContextPred* [4]	72.5±1.5	74.0±3.4	59.7±1.8	75.6±1.0	73.6±0.3	78.8±1.2	62.6±0.6	70.6±1.5	70.9
GraphMAE* [5]	76.3±2.4	82.3±1.2	60.3±1.1	77.2±1.0	75.5±0.6	83.1±0.9	64.1±0.3	72.0±0.6	73.9
Uni-Mol [44]	72.0±0.5	81.2±1.7	57.7±0.3	78.3±0.2	**78.9**±0.4	78.1±1.4	62.7±0.4	65.4±1.0	71.8
Mole-BERT* [31]	78.6±1.8	78.9±3.0	62.8±1.1	78.2±0.8	76.8±0.5	80.8±1.4	64.3±0.2	71.9±1.6	74.0
JOAO* [35]	76.9±0.7	66.6±3.1	60.4±1.5	76.9±0.7	74.8±0.6	73.2±1.6	62.8±0.7	66.4±1.0	71.1
GraphLoG* [45]	76.0±1.1	76.7±3.3	61.2±1.1	77.8±0.8	75.7±0.5	83.5±1.2	63.5±0.7	72.5±0.8	73.4
D-SLA* [30]	76.6±0.9	80.2±1.5	60.2±1.1	78.6±0.4	76.8±0.5	**83.8**±1.0	64.2±0.5	72.6±0.8	73.9
MoAMa w/o $\mathcal{L}_{\text{aux}}$	78.5±0.4	80.9±0.8	61.2±0.2	79.2±0.5	76.2±0.3	82.6±0.2	**64.6**±0.1	71.8±0.7	74.4
MoAMa	**80.0**±0.8	**85.3**±2.2	**64.6**±0.5	**79.3**±0.6	76.5±0.1	80.1±0.5	63.0±0.4	**72.8**±0.9	**75.3**

Table 2: Test RSME performance on three Open Graph Benchmark (OGB) molecular regression datasets. The best RMSE values for each dataset are in **bold**. Models with (*) use the same GNN architecture as MoAMa.

	ESOL	FreeSolv	Lipophilicity	**Avg**
No Pretrain*	1.605±0.08	3.589±0.33	0.831±0.02	2.009
MGSSL* [29]	1.438±0.12	2.840±0.19	0.869±0.06	1.716
MCM [43]	1.553±0.11	3.051±0.11	0.815±0.11	1.806
Grover [32]	3.107± 1.39	7.939±3.17	1.377±0.06	3.141
AttrMask* [4]	1.486±0.12	2.934±0.13	0.817±0.01	1.746
ContextPred* [4]	1.597±0.14	3.057±0.17	0.835±0.01	1.830
GraphMAE* [5]	1.510±0.09	3.060±0.15	0.852±0.02	1.807
Uni-Mol [44]	1.717±0.08	2.811±0.16	1.132±0.05	1.887
Mole-BERT* [31]	1.742±0.05	4.029±0.08	0.821±0.02	1.901
JOAO* [35]	1.640±0.09	3.467±0.06	1.040±0.05	2.163
GraphLoG* [45]	1.507±0.13	3.467±0.06	1.040±0.05	2.005
D-SLA* [30]	1.538±0.13	2.920±0.05	0.790±0.09	1.749
MoAMa w/o $\mathcal{L}_{\text{aux}}$	1.409±0.07	2.986±0.09	0.864±0.05	1.753
MoAMa	**1.394**±0.12	**2.700**±0.11	**0.772**±0.03	**1.622**

where $\mathbf{h}_{G_i}$ and $\mathbf{h}_{G_j}$ are the graph representation of G_i and G_j, respectively. The full pre-training loss is $\mathcal{L} = \beta\mathcal{L}_{\text{rec}} + (1-\beta)\mathcal{L}_{\text{aux}}$, where β is a hyperparameter to balance these two loss terms ($\beta = 0.5$).

6 Experiments

6.1 Experimental Settings

Datasets. Following the setting of previous studies [5, 30, 31], 2 million unlabeled molecules from the ZINC15 [46] was used to pre-train the GNN models. To evaluate the performance on downstream tasks, experiments were conducted across eleven classification/regression benchmark datasets from MoleculeNet [1].

Validation methods and evaluation metrics. In accordance with previous work, we adopt a scaffold splitting approach [4, 29] in which molecules are divided according to structures into train, validation, and test sets [1], using a 80:10:10 split for the three sets. We use the area under the ROC curve (AUC) for classification and RMSE for regression to evaluate the test performance of the best validation step during 10 independent runs.

Table 3: Contribution of the loss term $\mathcal{L}_{\text{aux}}$ to AttrMask and MoAMa.

	MUV	ClinTox	SIDER	HIV	Tox21	BACE	ToxCast	BBBP	**Avg**
$\mathcal{L}_{\text{aux}}$ Only	68.8±0.7	71.2±2.1	59.6±0.5	75.7±0.3	74.5±0.5	81.3±1.3	63.5±0.5	67.7±1.0	70.2
AttrMask	75.8 ±1.0	73.5±4.3	60.5±0.9	75.3±1.5	75.1±0.9	77.8±1.8	63.3±0.6	65.2±1.4	70.8
AttrMask + $\mathcal{L}_{\text{aux}}$	78.5±0.6	72.9±1.4	62.0±0.6	76.4±1.1	**76.5**±0.4	70.3±0.9	63.8±0.4	70.3±1.0	72.8
MoAMa w/o $\mathcal{L}_{\text{aux}}$	78.5±0.4	80.9±0.8	61.2±0.2	79.2±0.5	76.2±0.3	**82.6**±0.2	**64.6**±0.1	71.8±0.7	74.4
MoAMa	**80.0**±0.8	**85.3**±2.2	**64.6**±0.5	**79.3**±0.6	**76.5**±0.1	80.1±0.5	63.0±0.4	**72.8**±0.9	**75.3**

Table 4: Measurements of inter-motif knowledge transfer using pre-trained models. A higher ratio is preferred for the InfRatio measurements, and a lower score is preferred for the MRR measurements.

Model	**Avg** Test AUC	$\text{InfRatio}_{\text{node}}\uparrow$	$\text{InfRatio}_{\text{graph}}\uparrow$	$\text{MRR}_{\text{node}}\downarrow$	$\text{MRR}_{\text{graph}}\downarrow$	$\text{MRR}_{\text{motif}}\downarrow$
AttrMask	70.8	0.70	0.44	0.66	0.64	0.51
ContextPred	70.9	0.71	0.48	0.65	0.64	0.51
JOAO	71.1	0.87	0.55	0.59	0.57	0.49
MGSSL	72.3	0.60	0.38	0.73	0.73	0.60
GraphLoG	73.4	0.79	0.50	0.61	0.59	0.48
D-SLA	73.8	0.76	0.49	0.62	0.61	0.49
GraphMAE	73.9	0.76	0.48	0.64	0.61	0.49
Mole-BERT	74.0	0.71	0.47	0.66	0.65	0.54
MoAMa	**75.3**	**0.89**	**0.56**	**0.50**	**0.49**	**0.44**

Model configurations. For fair comparison with previous works, a five-layer Graph Isomorphism Network (GIN) with an embedding dimension of 300 was chosen for the GNN encoder. The READOUT strategy is mean pooling. During pre-training and fine-tuning, models were trained for 100 epochs using the Adam optimizer, a learning rate of 0.001, and a batch size of 32.

Baselines. There are two general types of baseline graph pre-training strategies that we evaluate our work against: **contrastive learning** tasks, such as D-SLA [30], GraphLoG [45], and JOAO [35], and **attribute reconstruction**, including Grover [32], AttrMask [4], ContextPred [4], GraphMAE [5], Mole-BERT [31], and Uni-Mol [44]. Additionally, we evaluate on the **motif-based pre-training** strategy MGSSL [29], which recurrently generates the motif tree for any molecule, and MCM [43], which uses a motif-based convolution module to generate embeddings.

6.2 Results

For the classification tasks, we report AUC-ROC of different graph pre-training methods in Table 1. MoAMa outperforms all baseline methods on five out of eight datasets. On average, MoAMa outperforms the best baseline method Mole-BERT [31] by 1.3% and the best contrastive learning methods D-SLA [30] by 1.4%. For the regression tasks, we report RMSE in Table 2. MoAMa outperforms all baselines on all three datasets. On average, MoAMa outperforms the best baseline MGSSL [29] by 0.094. It is interesting to note that MGSSL [29] and MCM [43], motif-based works, are two of the more competitive baselines, implying that training models with motif knowledge may lead to better downstream results for regression tasks.

Comparing MoAMa with and without the auxiliary loss $\mathcal{L}_{aux}$, we see that while MoAMa without $\mathcal{L}_{aux}$ shows competitive performance to most of the baselines, the complementary knowledge from the substructure-based contrastive task enhances the overall model performance, on average, by 0.9% on the classification tasks and 0.131 on the regression tasks. Additionally, when investigating the contribution of $\mathcal{L}_{aux}$ to AttrMask in Table 3, we see that the additionally loss term also complements the random masking with a 2.0% performance increase. However, even with $\mathcal{L}_{aux}$, AttrMask does not exceed the performance of both MoAMa with and without $\mathcal{L}_{aux}$.

6.3 Inter-motif Influence Analysis

In Table 4, we report the two InfRatio and three MRR measurements for our model and several baselines. A higher influence ratio indicates that inter-motif nodes have a greater effect on the target node. The relatively low values indicate that the intra-motif node influence is still highly important for pre-training, but our method demostrates the highest inter-motif knowledge transfer amongst the baselines. We see a small positive correlation between the average test AUC for each model and the

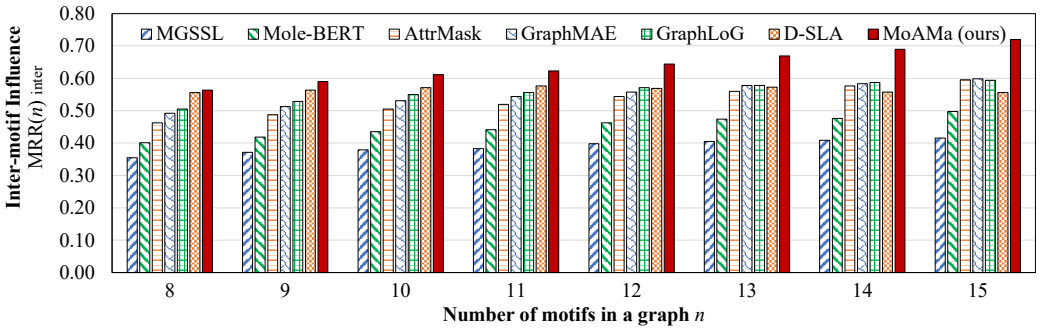


Figure 3: Inter-motif knowledge transfer score by motif count. A higher $\text{MRR}_{\text{inter}}^{(n)}$ score denotes greater inter-motif knowledge transfer.

InfRatio measurements, which supports our claim that greater inter-motif knowledge transfer leads to higher predictive performance. For the MRR measurements, our method boasts the lowest scores, which indicates less intra-motif knowledge dependence and greater inter-motif knowledge transfer.

For the sake of clear visualization, we define an inter-motif score which indicates inter-motif knowledge transfer according to the number of motifs n within a graph:

$$\text{MRR}_{\text{inter}}^{(n)} = 1 - \frac{1}{\sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}^{(n)}} |\mathcal{V}|} \sum_{(\mathcal{V},\mathcal{E})\in\mathcal{G}^{(n)}} \sum_{v\in\mathcal{V}} \frac{1}{\text{rank}_v}. \tag{15}$$

Figure 3 shows that our method outperforms the baselines in terms of inter-motif knowledge transfer as shown by the higher $\text{MRR}_{\text{inter}}^{(n)}$ across different motif counts. Additionally, the inter-motif knowledge transfer using our method becomes more pronounced on graphs with higher numbers of motifs.

Evaluation Complexity. In the worst-case, evaluation of inter-motif influence is computed between every pair of nodes within a molecule, causing an evaluation complexity of $O(n^2)$, for n is the number of nodes in graph G. However, GNN message passing is limited by the number of layers used, k. Therefore, the influence calculations only need to be performed on neighbors within a k-hop radius of each other. This means the time complexity of our evaluation is $O(n\bar{d}^k)$, where n is the number of nodes in the graph, k is the number of layers of our GNN ($k = 5$), and $\bar{d}$ is the average degree of a node. $\bar{d}^k \leq n$ as molecular graphs are sparse, so the evaluation is not as inefficient as $O(n^2)$.

7 Conclusions

In this work, we introduced a novel motif-aware attribute masking strategy for attribute reconstruction in graph model pre-training. This strategy outperformed existing random attribute masking methods and achieved competitive results with the state-of-the-art methods on eleven classification and regression datasets due to the explicit transfer of long-range inter-motif knowledge and intra-motif structural information. We quantitatively verify the increase in inter-motif knowledge transfer of our strategy over previous works using inter-motif node influence measurements and show strong correlations with our measurements and model accuracy.

While this work has demonstrated the importance of transferring motif-based knowledge through the graph pre-training framework, our strategy relies on the auxiliary loss to encode global structures and motif features. It would be compelling for future work to be able to encode global structure information using a motif-level message propagation method to capture long-distance motif dependencies, without relying on Tanimoto similarity.

Additionally, using a more general graph decomposition method would enable this work to expand to other graph applications, such as networks. It may then become possible to investigate node-level and edge-level tasks.

Acknowledgements

This work was supported by NSF IIS-2142827, IIS-2146761, IIS-2234058, CBET-2332270, and ONR N00014-22-1-2507.

References

[1] Zhenqin Wu, Bharath Ramsundar, Evan Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9, 03 2017. doi: 10.1039/C7SC02664A. 1, 2, 7

[2] Rees Chang, Yu-Xiong Wang, and Elif Ertekin. Towards overcoming data scarcity in materials science: unifying models and datasets with a mixture of experts framework, 2022. 1

[3] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*, 2021. 1

[4] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks, 2020. 1, 2, 3, 6, 7, 8, 13, 14

[5] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, C. Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. 2, 3, 6, 7, 8, 13, 14

[6] Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016. 1, 2

[7] Jun Xia, Yanqiao Zhu, Yuanqi Du, and Stan Z. Li. A survey of pretraining on graphs: Taxonomy, methods, and applications, 2022. 1

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 3, 14

[10] Phillip Pope, Soheil Kolouri, Mohammad Rostrami, Charles Martin, and Heiko Hoffmann. Discovering molecular functional groups using graph convolutional neural networks, 2019. 1

[11] Jean MJ Frechet. Functional polymers and dendrimers: reactivity, molecular architecture, and interfacial energy. *Science*, 263(5154):1710–1715, 1994. 1

[12] Merichel Plaza, Tania Pozzo, Jiayin Liu, Kazi Zubaida Gulshan Ara, Charlotta Turner, and Eva Nordberg Karlsson. Substituent effects on in vitro antioxidizing properties, stability, and solubility in flavonoids. *Journal of agricultural and food chemistry*, 62(15):3321–3333, 2014. 1

[13] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017. 1

[14] Zhanghao Wu, Paras Jain, Matthew A. Wright, Azalia Mirhoseini, Joseph E. Gonzalez, and Ion Stoica. Representing long-range context for graph neural networks with global attention, 2022. URL https://arxiv.org/abs/2201.08821. 1

[15] Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions, 2017. URL https://arxiv.org/abs/1706.08566.

[16] Marco Anselmi, Greg Slabaugh, Rachel Crespo-Otero, and Devis Di Tommaso. Molecular graph transformer: stepping beyond alignn into long-range interactions. *Digital Discovery*, 3 (5):1048–1057, 2024. 1

[17] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding, 2019. 2

[18] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks, 2020. 2, 3, 14

[19] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in*

Neural Information Processing Systems, volume 34, pages 13683–13694. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/71ddb91e8fa0541e426a54e538075a5a-Paper.pdf. 2

[20] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications, 2021. 2

[21] Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance, 2023. 2

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 2

[23] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders, 2022. 2

[24] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022. 2

[25] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration, 2019. 2

[26] Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. Pmi-masking: Principled masking of correlated spans, 2020.

[27] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans, 2020. 2

[28] Gang Liu, Eric Inae, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Data-centric learning from unlabeled graphs with diffusion model. *arXiv preprint arXiv:2303.10108*, 2023. 2, 3

[29] Zaixin Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Motif-based graph self-supervised learning for molecular property prediction. In *Neural Information Processing Systems*, 2021. 2, 3, 4, 7, 8

[30] Dongki Kim, Jinheon Baek, and Sung Ju Hwang. Graph self-supervised learning with accurate discrepancy learning. *ArXiv*, abs/2202.02989, 2022. 2, 7, 8

[31] Jun Xia, Chengshuai Zhao, Bozhen Hu, Zhangyang Gao, Cheng Tan, Yue Liu, Siyuan Li, and Stan Z. Li. Mole-BERT: Rethinking pre-training graph neural networks for molecules. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jevY-DtiZTR. 2, 3, 7, 8

[32] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. 3, 7, 8

[33] Mengying Sun, Jing Xing, Huijun Wang, Bin Chen, and Jiayu Zhou. Mocl: Data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3585–3594, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467186. URL https://doi.org/10.1145/3447548.3467186. 3

[34] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020. 3

[35] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021. 3, 7, 8

[36] Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark, 2023. 3

[37] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km. 3

[38] Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3, 2008. 4

[39] Zijie Geng, Shufang Xie, Yingce Xia, Lijun Wu, Tao Qin, Jie Wang, Yongdong Zhang, Feng Wu, and Tie-Yan Liu. De novo molecular generation via connection-aware motif mining. *arXiv preprint arXiv:2302.01129*, 2023. 4

[40] Nick Craswell. *Mean Reciprocal Rank*, pages 1703–1703. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_488. URL https://doi.org/10.1007/978-0-387-39940-9_488. 5

[41] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7, 2015. 6

[42] Austin Atsango, Nathaniel L. Diamant, Ziqing Lu, Tommaso Biancalani, Gabriele Scalia, and Kangway V. Chuang. A 3d-shape similarity-based contrastive approach to molecular representation learning, 2022. 6

[43] Yifei Wang, Shiyang Chen, Guobin Chen, Ethan Shurberg, Hang Liu, and Pengyu Hong. Motif-based graph representation learning with application to chemical molecules, 2022. 7, 8

[44] Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:259298651. 7, 8

[45] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11548–11558. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xu21g.html. 7, 8

[46] T. Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55:2324 – 2337, 2015. 7

[47] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation, 2015. 14

[48] Chaoning Zhang, Chenshuang Zhang, Junha Song, John Seon Keun Yi, Kang Zhang, and In So Kweon. A survey on masked autoencoder for self-supervised learning in vision and beyond, 2022. 14

[49] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 889–898, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3132967. URL https://doi.org/10.1145/3132847.3132967. 14

[50] Amin Salehi and Hasan Davulcu. Graph attention auto-encoders, 2019.

[51] Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6519–6528, 2019. 14

Table 5: Strategy design for motif-aware attribute masking: (1) masking distribution, (2) reconstruction target, (3) reconstruction loss, and (4) decoder model. The chosen design is highlighted.

Design Space		MUV	ClinTox	SIDER	HIV	Tox21	BACE	ToxCast	BBBP	**Avg**
(1)	100% Motif Coverage	80.0±0.8	85.3±2.2	64.6±0.5	79.3±0.6	76.5±0.1	80.1±0.5	63.0±0.4	72.8±0.9	**75.3**
	75% Node-wise	74.9±1.1	82.3±0.4	60.1±0.3	78.8±0.9	76.1±0.1	82.3±0.4	63.4±0.1	72.1±1.0	73.7
	75% Element-wise	74.8±0.7	84.9±1.0	58.7±0.1	79.7±0.7	75.6±0.1	85.7±0.4	63.4±0.2	72.6±0.4	74.4
	50% Node-wise	76.6±1.2	86.4±0.6	58.3±0.1	78.1±0.3	75.1±0.2	81.9±0.3	64.6±0.1	72.7±0.1	74.2
	50% Element-wise	73.9±0.2	71.2±4.0	61.2±0.4	77.5±0.8	74.9±0.4	81.1±0.7	62.5±0.1	70.6±1.8	71.6
	25% Node-wise	76.6±1.5	86.3±0.7	62.4±0.2	78.4±0.2	75.9±0.2	81.8±0.1	65.1±0.1	74.7±0.2	75.1
	25% Element-wise	75.2±1.5	82.1±0.4	58.3±0.1	77.8±1.5	75.5±0.2	81.5±0.2	63.1±0.1	71.6±0.3	73.1
(2)	Atom Type	80.0±0.8	85.3±2.2	64.6±0.5	79.3±0.6	76.5±0.1	80.1±0.5	63.0±0.4	72.8±0.9	**75.3**
	Chirality	76.3±1.8	75.1±0.9	59.8±0.5	77.9±0.1	76.6±0.1	79.8±0.5	63.8±0.2	73.8±0.7	72.9
	Both w/ one decoder	76.2±1.4	74.4±1.1	62.4±0.9	78.2±1.1	75.5±0.6	82.1±0.4	64.3±0.2	72.9±0.2	73.3
	Both w/ two decoders	75.9±0.9	81.5±0.1	60.5±0.1	78.5±0.9	75.8±0.2	82.0±1.0	63.7±0.3	73.4±0.3	73.9
(3)	Scaled Cosine Error	80.0±0.8	85.3±2.2	64.6±0.5	79.3±0.6	76.5±0.1	80.1±0.5	63.0±0.4	72.8±0.9	**75.3**
	Cross Entropy	78.8±1.1	84.5±0.7	65.4±0.2	78.6±0.4	76.3±0.1	82.4±0.2	62.9±0.5	72.3±0.2	75.1
	Mean Squared Error	80.0±0.5	84.1±1.4	64.6±0.5	78.3±0.4	76.8±0.2	80.5±0.6	62.8±0.3	71.8±0.6	74.9
(4)	GNN decoder	80.0±0.8	85.3±2.2	64.6±0.5	79.3±0.6	76.5±0.1	80.1±0.5	63.0±0.4	72.8±0.9	**75.3**
	MLP decoder	78.8±0.5	85.2±0.1	65.5±0.3	78.1±0.6	76.2±0.2	82.1±0.6	62.8±0.8	71.7±0.4	75.1

A Appendix

A.1 Inter-motif Influence

The points in Fig. 2 represent pre-trained models using an attribute reconstruction strategy. Pre-training strategies include AttrMask, ContextPred, GraphMAE, Mole-BERT, and MoAMa (denoted using ◆). Several points may represent the same pre-training strategy but with changes to the masking rate of the model. Additional points for AttrMask and Mole-BERT were generated, using mask rates of 15% (default hyperparameter), 20%, 25%, and 30% (MoAMa's upper bound). The classification tasks are the same as those used for evaluation.

A.2 Design Space of the Attribute Masking Strategy

The design space of the motif-aware node attribute masking includes the following four parts:

Masking distribution. We investigate the influence of masking distribution to the masking strategy using two factors to control the distribution of masked attributes:

- Percentage of nodes within a motif selected for masking: we propose to mask nodes from the selected motifs at different percentages. The percentage indicates the strength of the masked domain knowledge, which affects the hardness of the pre-training task of the attribute reconstruction.
- Dimension of the attributes: We propose to conduct either node-wise or element-wise (dimension-wise) masking. Element-wise masking selects different nodes for masking in different dimensions according to the percentage, while node-wise masking selects different nodes for all-dimensional attribute masking in different motifs.

For motif-aware masking, there is the choice of masking the features of all nodes within the motif or choosing to only mask the features of a percentage of nodes within each sampled motif. For our study, we choose a motif coverage parameter to decide what percentage of nodes within each motif to mask, ranging from 25%, 50%, 75%, or 100%.

Furthermore, the masking strategy utilized by previous work performs node-wise masking [4, 5], where all features of a node are masked. An alternative strategy may be element-wise masking, where masked elements are chosen over all feature dimensions and implies that not all features of a node may necessarily be masked. Note that 100% masking will behave the exact same as node-wise masking, as 100% of nodes within a motif will have each feature masked.

We provide the predictive performance within Table 5. The predictive performance for the node-wise masking outperforms the element-wise masking for both 25% and 50% node coverage. At 75% coverage, element-wise masking outperforms node-wise. However, the full coverage masking strategy

outperforms all other masking strategies, due to the hardness of the pre-training task, which enables greater transfer of inter-motif knowledge.

Reconstruction target. Existing molecular graph pre-training methods heavily rely on two atom attributes: atom type and chirality. Therefore, the reconstructive task could include one or both attributes using one or two different decoders. The choice of attributes to reconstruct for GNNs towards molecular property prediction has traditionally been atom type [4, 5]. We verify the choice of reconstruction attributes by comparing the performance of the baseline model against models trained by reconstructing only chirality, both atom type and chirality using two separate decoders, or both properties using one unified decoder. From Table 5, we note that predicting solely atom type yields the best pre-training results. The second best strategy was to predict both atom type and chirality using two decoders. In this case, the loss of the two decoders are independent, leading to the conclusion that the chirality prediction task is ill-suited to be the pre-training task. Because choice of chirality is limited to four extremely imbalanced outputs, the useful transferable knowledge may be significantly lesser than that of atom prediction, which, for the ZINC15 dataset, has nine types.

Reconstruction loss. For the pretraining task, we have three choices of error functions to calculate training loss. A standard error function used for masked autoencoders within computer vision [9, 47, 48] is the cross-entropy loss, whereas previous GNN solutions utilize mean squared error (MSE) [18, 49–51]. GraphMAE [5] proposed that cosine error could mitigate sensitivity and selectivity issues:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\mathcal{V}_{[\text{MASK}]}|} \sum_{v \in \mathcal{V}_{[\text{MASK}]}} (1 - \frac{\mathbf{X}_v^T \mathbf{H}_v}{||\mathbf{X}_v|| \cdot ||\mathbf{H}_v||})^\gamma, \gamma \geq 1. \tag{16}$$

This equation is called the scaled cosine error (SCE). $\mathbf{H}$ are the reconstructed features, $\mathbf{X}$ are the ground-truth node features, and γ is a scaling factor ($\gamma = 1$) We investigate the effect these different error functions have on downstream predictive performance in Table 5 and find that SCE outperforms CE and MSE, in accordance with previous work.

Decoder model. The decoder trained via Eq. (13) could be a GNN or a MLP. Although the GNN decoder might be powerful [5], we are curious if the MLP delivers a comparable or better performance with higher efficiency.

We follow the GNN decoder settings from previous work [5] to conduct our study to determine which decoder leads to better downstream predictive performance. In Table 5, we show that our method outperforms the MLP-decoder strategy, which support previous work that show MLP-based decoders lead to reduced model expressiveness because of the inability of MLPs to utilize the high number of embedded features [5].

A.3 Classification Tasks

Dataset statistics for the classification tasks are given in Table 6.

Dataset	# of Graphs	Avg. Nodes	Avg. Edges
MUV	93,087	24.2	26.3
ClinTox	1,477	26.2	55.8
SIDER	1,427	33.6	70.7
HIV	41,127	25.5	54.9
Tox21	7,831	18.6	38.6
BACE	1,513	34.1	73.7
ToxCast	8,576	18.8	38.5
BBBP	2,039	24.1	51.9

Table 6: Statistics of eight datasets for graph classification.

A.4 Regression Tasks

Dataset statistics for the regression tasks are given in Table 7.

Dataset	# of Graphs	Avg. Nodes	Avg. Edges
ESOL	1,128	13.3	27.4
FreeSolv	642	8.7	16.8
Lipophilicity	4,200	27.0	59.0

Table 7: Statistics of three datasets for graph regression.