MOTIF-DRIVEN CONTRASTIVE LEARNING OF GRAPH REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph motifs are significant subgraph patterns occurring frequently in graphs, and they play important roles in representing the whole graph characteristics. For example, in chemical domain, functional groups are motifs that can determine molecule properties. Mining and utilizing motifs, however, is a non-trivial task for large graph datasets. Traditional motif discovery approaches mostly rely on exact counting or statistical estimation, which are hard to scale for a large number of graphs with continuous and high-dimension features. In light of the significance and challenges of motif mining, we propose *MICRO-Graph*: a framework for MotIf-driven Contrastive leaRning Of Graph representations to: 1) pre-train Graph Neural Networks (GNNs) in a self-supervised manner to automatically extract graph motifs from large graph datasets; 2) leverage learned motifs to guide the contrastive learning of graph representations, which further benefit various graph downstream tasks. Specifically, given a graph dataset, a motif learner cluster similar and significant subgraphs into corresponding motif slots. Based on the learned motifs, a motif-guided subgraph segmenter is trained to generate more informative subgraphs, which are used to conduct graph-to-subgraph contrastive learning of GNNs. By pre-training on ogbn-molhiv molecule dataset with our proposed *MICRO-Graph*, the pre-trained GNN model can enhance various chemical property prediction downstream tasks with scarce label by 2.8%, and significantly higher than other state-of-the-art self-supervised learning baselines.

1 INTRODUCTION

Graph-structured data, such as molecules and social networks, is ubiquitous in many scientific research areas and real-world applications. To represent graph characteristics, graph motifs were proposed in Milo et al. (2002) as significant subgraph patterns occurring frequently in graphs and uncovering graph structural principles. Functional groups are important motifs that can determine molecule properties. For example, for chemical compounds, hydroxide ($-OH$) usually implies higher water solubility, and for proteins, Zif268 can mediate protein-protein interactions in sequence-specific DNA-binding proteins. (Pabo et al., 2001).

Researchers have been studying graph motifs for years. On one hand, utilizing meaningful motifs can benefit many important applications like quantum chemistry and drug discovery (Ramsundar et al., 2019). On the other hand, extracting motifs from large graph datasets is still a challenging question. Traditional motif discovery approaches (Milo et al., 2002; Kashtan et al., 2004; Schreiber & Schwöbbermeyer, 2005; Chen et al., 2006; Wernicke, 2006) mostly rely on discrete counting or statistical estimation, and thus are hard to generalize to large-scale graph datasets with continuous and high-dimension features, as often the case in real-world applications.

Recently, Graph Neural Networks (GNNs) have shown great expressive power for learning graph representations without explicit structural feature engineering (Kipf & Welling, 2016; Hamilton et al., 2017; Veličković et al., 2017; Xu et al., 2018). In addition, GNNs can be trained in a self-supervised manner without human annotations to capture important graph structural and semantic properties (Veličković et al., 2018; Hu et al., 2020c; Qiu et al., 2020; Bai et al., 2019; Navarin et al., 2018; Wang et al., 2020; Sun et al., 2019; Hu et al., 2020b). This motivates us to rethink about motifs as more general representations than exact structure matches and ask the following research questions:

- Can we use GNNs to automatically extract graph motifs from large graph datasets?
- Can we leverage the learned graph motif to benefit self-supervised GNN learning?

In this paper, we propose *MICRO-Graph*: a framework for MotIf-driven Contrastive leaRning Of Graph representations. The key idea of this framework is to learn graph motifs as prototypical cluster centers of subgraph embeddings encoded by GNNs. In this way, the discrete counting problem is transfered to a fully-differentiable framework that can generalize to large-scale graph datasets with continuous and high-dimensional features. In addition, the learned motif can help generate more informative subgraph for graph-to-subgraph contrastive learning. The motif learning and contrastive learning are mutually reinforced, and eventually help pre-train a more generalizable GNN encoder.

For motif learning, given a graph dataset, a motif-guided subgraph segmenter generates subgraph samples for each graph, and the GNN encoder turns these subgraphs into d-dimensional vector representations. We learn graph motifs also as d-dimensional vectors through clustering, where K prototypical cluster centers are kept in an embedding table as representations of motifs. Similar and significant subgraphs are assigned to the same motif, and thus moved to be closer to the corresponding motif representation. We train our model in an Expectation-Maximization (EM) fashion to update both the motif assignment of each subgraph and the representations of motifs.

For leveraging learned motifs, we propose a novel graph-to-subgraph contrastive learning framework for GNN pre-training. One of the key components for a successful contrastive learning is to generate semantically meaningful and diverse views of each data instance. For example, a continuous span within a sentence (Joshi et al., 2020) or a random crop of a image (Chen et al., 2020). Previous approaches, such as Deep Graph Infomax and InfoGraph leverage node-to-graph views for contrastive prediction, which is not sufficient enough to capture the subgraph information. To address this issue, we propose to learn graph-to-subgraph contrastive. As motifs can represent the key graph property by its nature, we propose to leverage the learned motifs to generate more informative subgraphs, and benefit GNN contrastive learning. For example, alpha helix and beta sheet can come together as a simple $\beta\beta\alpha$ fold to form a zinc finger protein, and have unique properties. By learning such subgraph co-occurrence via contrastive learning, the pre-trained GNN should capture more subtle structural semantics of graph.

After pre-training on ogbg-molhiv dataset using our proposed *MICRO-Graph*, the framework can successfully learn meaningful motifs, including Benzene ring, Nitro and Acetate, etc. Meanwhile, the pre-trained GNN encoder can be used to fine-tune on seven chemical property prediction benchmarks, yielding 2.8% average improvement over non-pretrained GNNs and outperforming other self-supervised pre-training baselines. Extensive ablation studies further show that the learned motifs are indeed significant for the contrastive learning framework.

2 Related Work

The goal of self-supervised learning is to train a neural network model to capture significant characteristics of data without human annotations. This paper studies whether we can use such framework to automatically extract graph motifs, one of the significant structural patterns in graph, and leverage the learned motifs to benefit self-supervised learning. In the following, we first review some preliminaries about graph motifs and the challenge for motif mining, and then review approaches of pre-training graph neural networks in a self-supervised manner.

Graph motifs are building blocks of complex graphs. They reveal the interconnections of graphs and represent graph characteristics. Mining motifs can benefit different tasks, from exploratory analysis to graph transfer learning (Henderson et al., 2012). Motif discovery has been studied for many years, and various algorithms have been proposed. There are generally two categories for automatically extracting motifs from graph data, either exact counting as in Milo et al. (2002); Kashtan et al. (2004); Schreiber & Schwöbbermeyer (2005); Chen et al. (2006), or sampling and statistical estimation as in Wernicke (2006). However, both approaches can not scale to large graph datasets with continuous and high-dimension features, which is often the case in real-world applications. In this paper, we proposes to turn the discrete motif mining problem into differentiable cluster learning, and adopt a self-supervised GNNs learning framework to automatically extract important graph motifs. Such method can be generalized to large-scale graph datasets.

Contrastive learning is one of the state-of-the-art self-supervised representation learning algorithms that achieves great results for visual representation learning (Chen et al., 2020; He et al., 2019). Contrastive learning force that embeddings of views generated from the same data (e.g. a image crops) instance to be closer to each other, while those from different instances apart. One of the key component in contrastive learning is to generate informative and diverse views from each data instance. In computer vision, researchers adopt various image augmentation techniques, including crop, color distortion, Gaussian blurs, to generate such views. However, when in comes to graph domain, constructing informative view of graph, e.g., subgraph, is a challenging task. In our framework, we utilize the learned motifs, which are significant subgraph patterns, to guide constructing informative views from each graph, and conduct graph-to-subgraph contrastive learning.

Self-supervised learning for GNNs also draws many attention recently. For graphs, representations can be at different levels, e.g. node level and (sub)graph level. Veličković et al. (2018); Hu et al. (2020c); Qiu et al. (2020) mainly focus on node-level representation learning in a single large graph, as opposed to the focus of this paper, representations of the whole graphs. Hu et al. (2020b) provides a systematic analysis of pre-training strategies on graphs for both node-level and graph-level representations. However, only the node-level learning is self-supervised, and annotated labels are utilized to perform supervised learning at the graph level. For graph level self-supervised representation learning, Sun et al. (2019) proposed a contrastive framework InfoGraph to maximize the mutual information between the representations of the whole graphs and node representations. The views they utilize are each node in the graph with the whole graph, which capture more subtle subgraph semantics. To alleviate this issue, we propose a novel graph-to-subgraph view construction, and the subgraph generation is guided by learned motifs.

3 METHODOLOGY

The main goal of this paper is to train a GNN encoder that can automatically extract graph motifs, i.e. significant subgraph patterns. To avoid combinatorial problem of discrete graph structure, we propose to solve it by a differentiable clustering learning via self-supervised GNN learning. In this section, we first formalize the differentiable motif learning problem, and then introduce the overall framework of *MICRO-Graph* to extract graph motifs via clustering, and finally introduce how the learned motif can benefit graph-to-subgraph contrastive learning.

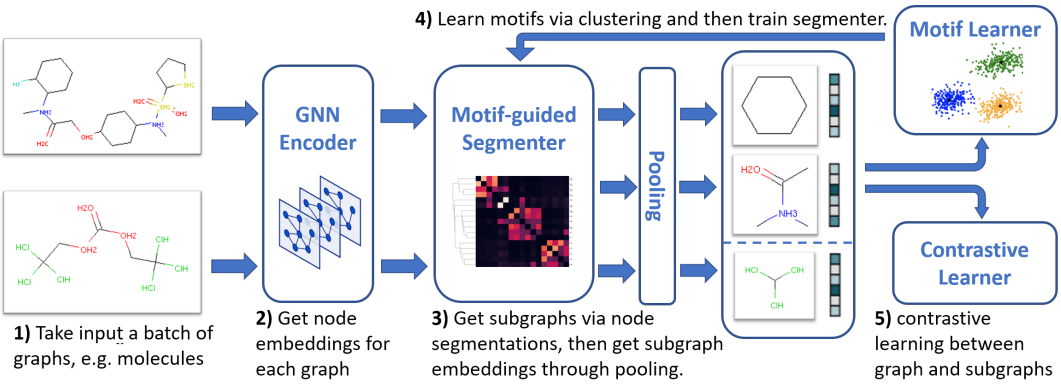

Figure 1: Overall framework of *MICRO-Graph*, which can train GNNs in a self-supervised manner to automatically extract motifs, and leverage the motif to generate more informative subgraphs for graph-to-subgraph contrastive learning.

3.1 THE OVERALL FRAMEWORK OF *MICRO-Graph*

Given a dataset with M graphs $\mathbb{G} = \{\mathcal{G}_1, ..., \mathcal{G}_M\}$, the differentiable clustering learning problem is to learn a GNN-based graph encoder $\boldsymbol{E}(\cdot)$ as well as a motif embedding table $\{\boldsymbol{m_1}, ..., \boldsymbol{m_k}\}$, each slot of which represents a motif vector that is a cluster center of frequently occurred subgraph

embeddings encoded by $\boldsymbol{E}(\cdot)$. To tackle this problem, our *MICRO-Graph* consists of three modules: 1) a motif-guided segmenter for extracting important subgraphs; 2) a motif-learning module for conducting clustering on sampled subgraphs and extracting motifs; 3) a constrastive learning module based on the learned motifs.

3.2 MOTIF LEARNER VIA EM CLUSTERING

We first introduce the *Motif Learner* that uses EM clustering to learn motifs. Utilizing the *Motif-guided Subgraph Segmenter*, we can extract M of subgraphs $\{g_j\}_{j=1}^{M}$ from a batch of graphs in the dataset. When sampling a subgraph g_j from a graph $\mathcal{G}_i$, we first get the node embeddings of this graph, and apply the subgraph index as a mask to its subgraph embedding, i.e., $\boldsymbol{e}_j = \frac{1}{|g_j|}\sum_{s\in g_j} E(\mathcal{G}_i)[s]$. Afterwards, for each subgraph j, we calculate the cosine similarity $\boldsymbol{s}_{j,k}$ between its embedding with the K motif vectors $\{\boldsymbol{m_1}, ..., \boldsymbol{m_k}\}$ as $\boldsymbol{s}_{j,k} = \phi(\boldsymbol{e}_j)^T\phi(\boldsymbol{m}_k)$.[1] By merging them together, we get the subgraph-motif similarity matrix $\boldsymbol{S} = [\boldsymbol{S_1}, ..., \boldsymbol{S_N}]$.

E-Step. The goal of cluster assignment is to find a probability matrix $\boldsymbol{Q} = [\boldsymbol{q_1}, ..., \boldsymbol{q_N}]$, where the j-th column $\boldsymbol{q}_j$ is a K-dimensional probability vector represents which motif the j-th subgraph is assigned to. The vanilla cluster assignment probability objective to be maximized is $Tr(\boldsymbol{Q}^T\boldsymbol{S})$. However, directly solving this optimization problem over all possible $\boldsymbol{Q}$ can lead to cluster degenerate solutions, where all representations are collapsed to a single cluster center. To avoid this issue, we follow YM. et al. (2020) to introduce an entropy term and an equal-size constraint on $\boldsymbol{Q}$ to make all the clusters to have similar sizes. Our final objective is:

$$\max_{\boldsymbol{Q}\in\mathcal{Q}} Tr(\boldsymbol{Q}^T\boldsymbol{S}) + \frac{1}{\lambda}H(\boldsymbol{Q}) \tag{1}$$

where $H(\boldsymbol{Q}) = -\sum_{i,j} Q_{i,j}\log Q_{i,j}$ is the entropy function, and the constraint set $\mathcal{Q}$ requires the marginal projection of $\boldsymbol{Q}$ onto its columns and rows to be uniform.

$$\mathcal{Q} = \{\boldsymbol{Q} \in \mathbb{R}_+^{K,N} | \boldsymbol{Q}\boldsymbol{1_N} = \frac{\boldsymbol{1_K}}{K}, \boldsymbol{Q}\boldsymbol{1_K} = \frac{\boldsymbol{1_N}}{N}\} \tag{2}$$

where $\boldsymbol{1_N}$ and $\boldsymbol{1_K}$ are all one vectors. This specific constraint optimization problem turns out to be an optimal transportation problem with a closed-form solution

$$\boldsymbol{Q}^* = diag(\boldsymbol{u}) \cdot \exp(\lambda\boldsymbol{Q}^T\boldsymbol{Z}) \cdot diag(\boldsymbol{v}) \tag{3}$$

Here $\boldsymbol{u}$ and $\boldsymbol{v}$ are normalization vectors and can be solved using a fast Sinkhorn-Knopp algorithm. (derivations in Cuturi (2013)).

M-Step. With this cluster assignment matrix $\boldsymbol{Q}$, we can update our representations as supervised learning. Specifically, we predict the probability of a subgraph embedding belonging to each cluster, and we optimize the cross entropy loss using values in $\boldsymbol{Q}$ as labels. The loss function is

$$\mathcal{L}_m = -\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{K} Q_{j,k}\log \mathrm{softmax}_j(S)_{j,k} \tag{4}$$

3.3 MOTIF-GUIDED SUBGRAPH SEGMENTER

Sampling informative subgraphs is a very crucial component for both motif learning and contrastive learning in *MICRO-Graph*. Traditional heuristic approaches such as random walk and k-hop sampling cannot guarantee to generate subgraph which is semantically reasonable and informative. For example, for molecule graphs, a subgraph produced by random walk might not meet the molecule rules.

Since motifs are by nature informative subgraph patterns, we propose to leverage the learned graph motifs to design the *Motif-guided subgraph segmenter*, such that it can learn to segment a given graph into several subgraphs that are close to some motif slots.

Subgraph Sampling via Node Segmentation. Given a graph $\mathcal{G}_i$ with n nodes, we first use the GNN encoder $\boldsymbol{E}(\cdot)$ to get all the node embeddings $\boldsymbol{n}_i = \boldsymbol{E}(\mathcal{G}_i)$, then we compute pairwise cosine

[1] For notation simplicity, we denote a L-2 normalization operator $\phi(\cdot)$ such that $\phi(x) = x \,/\, \|x\|_2$.

similarity between these node embeddings followed by row-wise softmax normalization to get a node affinity matrix $A \in \mathbb{R}^{N,n,n}$, where $A_{i,s,t}$ indicates the affinity score between node s and node t in graph i.

$$A_{i,s,t} = \mathrm{softmax}_s\big(\phi(\boldsymbol{n}_{i,s}^T)\phi(\boldsymbol{n}_{i,t})\big) \tag{5}$$

Afterwards, we apply spectral clustering based on the affinity matrix A_i for each graph $\mathcal{G}_i$ to segment nodes into different groups, within which the connected components that are larger than three nodes are collected as our sampled subgraphs $\{g_{i,j}\}_{j=1}^M$.

Motif-guided Training. To train the sampler so that the segmented subgraphs are close to some motifs, we use the subgraph-motif similarity matrix $\boldsymbol{S}$ as a learning signal to guide the sampler learning. For all those subgraphs that have a similarity score higher than threshold η to any motif, we force to maximize the affinity value among all the nodes within this subgraph.

$$\mathcal{L}_s = -\frac{1}{N}\sum_{i=1}^{N}\sum_{(s,t)\in g_{i,j}} A_{i,s,t}\cdot \mathbf{1}\{\exists k \mid \ S_{j,k} > \eta\} \tag{6}$$

By optimizing this loss, during the next sampling round, nodes produced motif-like subgraphs are more likely to come together as a sampled subgraph, which leads to more subgraph samples align with the motifs.

3.4 Contrastive Learner between graphs and subgraphs

To better conduct motif learning, we require an expressive GNN-based graph encoder $\boldsymbol{E}(\cdot)$ to capture the graph properties. We follow the contrastive learning framework, which is one of the state-of-the-art self-supervised learning method, to guide the GNN learning.

One of the key component in contrastive learning is to select informative and diverse views of the data instances. Previous methods either utilize node or the whole graph as the views, which neglect the minor substructure information. To alleviate this issue, we propose to use the subgraph generated by our *Motif-guided subgraph segmenter* as one view of the graph, and the whole graph as another. For each graph $\mathcal{G}_i$ within a batch of B graphs, we use GNN encoder to get node embeddings and average them to get the whole graph embedding as: $\boldsymbol{h}_i = \frac{1}{|\mathcal{G}_i|}\sum_{s\in\mathcal{G}_i} E(\mathcal{G}_i)[s]$. Then we consider the pairwise graph-to-subgraph similarity matrix $\boldsymbol{W} \in \mathbb{R}^{B,N}$, where

$$W_{i,j} = \mathrm{softmax}\big(\phi(\boldsymbol{g}_i^T)\phi(\boldsymbol{s}_j)\big) \tag{7}$$

For a particular whole graph, we consider all subgraphs sampled from this graph as positive pairs to it, and subgraphs sampled from other graphs as negative pairs. Thus, we can write down the objective function as:

$$\mathcal{L}_c = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{N}\log W_{i,j}\cdot \mathbf{1}\{\mathcal{S}_j \in \mathcal{G}_i\} \tag{8}$$

We show the main difference for generating graph views of our framework with existing methods in Figure 2. Most existing methods follow either node-to-node or node-to-graph view prediction. As is discussed in Chen et al. (2020), in computer vision domain, a more flexible view construction is to leverage random cropping and get a arbitrary-size image snippet. Our framework that utilizes subgraph-to-graph view prediction is similar to this intuition, with which the contrastive prediction task can capture higher-level structural patterns, and shows to enhance the self-supervised learning performance. Note that the subgraphs we utilize is generated by our *Motif-guided Subgraph Segmenter*. In the experiments, we systematically study the influence of subgraph sampling methods, and experiments show that simply utilizing heuristic sampling approaches leads to really bad generalization performance. This shows the significance of leveraging the learned graph motif to generate more informative and diverse view of graph-structured data.

Overall, by optimizing all the loss $\mathcal{L}_m + \mathcal{L}_s + \mathcal{L}_c$, our *MICRO-Graph* can simultaneously train a expressive GNN encoder and learn the underlying motifs of the given graph dataset. Intuitively, the motif learning and GNN learning is mutually reinforced: a better GNN encoder can help get more accurate subgraph embeddings and thus help motif mining, while a better motif can help generate more informative graph-to-subgraph views and help contrastive learning.

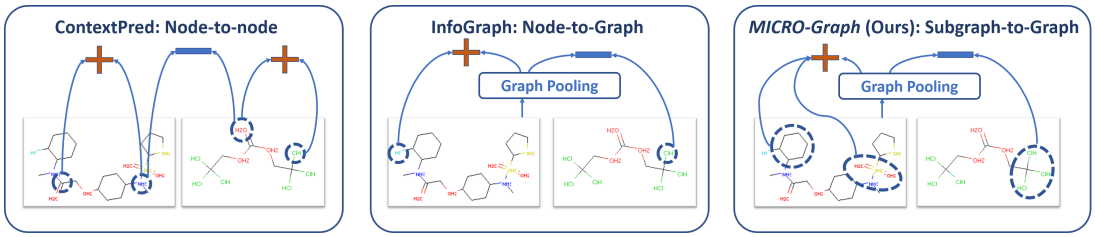

Figure 2: Difference ways to generate informative and diverse views for graph-data. Context Prediction only adopts node as views, and wish each node has similar embedding with its neighborhood while dissimilar to others; Deep Graph Infomax (DGI) and InfoGraph also consider the whole graph as another view, and thus learn via node-to-graph contrastive. Our method consider to use subgraph as a high-level views of each graph to conduct subgraph-to-graph contrastive learning, and thus the GNN can capture higher-order and global structure information.

4 EXPERIMENTS

We evaluate the effectiveness of *MICRO-Graph* from two perspective: 1) Whether the self-supervised framework can learn better GNNs to generalize well on graph classification tasks; 2) Whether the learned motifs are reasonable and can truly benefit contrastive learning.

4.1 EXPERIMENT SETUP

We mainly focus on chemical property prediction task. Specifically, we pre-train GNNs using *MICRO-Graph* on the ogbg-molhiv dataset from Open Graph Benchmark (OGB) (Hu et al., 2020a). This dataset contains 40K chemical compounds with average graph size 25. We test our pretrained model on other smaller molecule-related graph classification datasets from OGB. For more details of the datasets, please see Appendix C.

4.1.1 EVALUATION PROTOCOLS

After we pre-train a GNN encoder on ogbg-molhiv dataset, we evaluate its effectiveness using the following two evaluation protocols:

Transfer Learning Setting: we fine-tune the pre-trained GNN model with a small portion of labels of the downstream task. We adopt the same train-test and model selection procedure as in Yanardag & Vishwanathan (2015); Zhang et al. (2018); Xu et al. (2018), where we perform 10-fold cross-validation and select the epoch with the best cross-validation accuracy averaged over the 10 folds. We follow the OGB official setting to use the ROC-AUC as our metric.

Feature Extraction Setting: we fix the pre-trained GNN, use it as feature extractor to get graph representations of all the data in downstream tasks, and apply linear classifiers to train on top of them. We mainly follow the experimental settings in Sun et al. (2019). For datasets with multiple tasks, we report the averaged classification accuracy over all tasks.

4.1.2 BASELINES

For transfer learning setting, we adopt non-pretrain and direct supervised learning as the baseline. We also use three state-of-the-art GNN Self-Supervised Learning (SSL) methods as baselines:

InfoGraph (Sun et al., 2019) maximizes the mutual information between the representations of the whole graphs and the representations of its substructures at different granularity.

Context prediction (Hu et al., 2020b) uses subgraphs to predict their surrounding graph structures, so that nodes appearing in similar structural contexts will be mapped to nearby representations.

GPT-GNN (Hu et al., 2020c) predicts masked edges and masked node attributes. The edge prediction makes node representations to be close when there are edges between them. The attribute prediction captures how node attributes are distributed over all graphs.

SSL methods	bace	bbbp	clintox	hiv	sider	tox21	toxcast	Average
Non-Pretrain	72.80 ± 2.12	82.13 ± 1.69	74.98 ± 3.59	73.38 ± 0.92	55.65 ± 1.35	76.10 ± 0.58	63.34 ± 0.75	71.19
ContextPred	73.02 ± 2.59	80.94 ± 2.55	74.57 ± 3.05	73.85 ± 1.38	54.15 ± 1.54	74.85 ± 1.28	63.19 ± 0.94	70.65 (-0.7%)
InfoGraph	76.09 ± 1.63	80.38 ± 1.19	$\mathbf{78.36 \pm 4.04}$	72.59 ± 0.97	56.88 ± 1.80	76.12 ± 1.11	64.40 ± 0.84	72.11 (+1.3%)
GPT-GNN	75.56 ± 2.49	83.35 ± 1.70	74.84 ± 3.45	74.82 ± 0.99	55.59 ± 1.58	76.34 ± 0.68	64.76 ± 0.62	72.18 (+1.4%)
MICRO-Graph	$\mathbf{76.16 \pm 2.51}$	$\mathbf{83.78 \pm 1.77}$	77.50 ± 3.35	$\mathbf{75.51 \pm 0.67}$	$\mathbf{57.28 \pm 1.09}$	$\mathbf{76.68 \pm 0.36}$	$\mathbf{65.42 \pm 0.62}$	**73.19 (+2.8%)**

Table 1: Transfer learning performance (ROC-AUC) of MICRO-Graph compared with other self-supervised learning (SSL) baselines on molecule property prediction benchmarks. Pre-train GNNs on ogbg-molhiv dataset, fine-tune the pre-trained model on each downstream task for 10 times.

SSL methods	bace	bbbp	clintox	hiv	sider	tox21	toxcast	Average
ContextPred	57.13 ± 0.55	80.58 ± 0.21	$\mathbf{93.03 \pm 0.00}$	$\mathbf{96.49 \pm 0.00}$	75.35 ± 0.08	92.21 ± 0.01	83.68 ± 0.02	82.64
InfoGraph	67.51 ± 1.83	82.08 ± 1.13	90.92 ± 0.40	93.24 ± 1.92	68.80 ± 0.52	89.60 ± 0.37	80.19 ± 0.06	81.76
GPT-GNN	59.33 ± 0.26	73.82 ± 2.06	93.01 ± 0.07	94.16 ± 3.06	70.86 ± 0.34	88.60 ± 0.26	81.20 ± 0.12	80.14
MICRO-Graph	$\mathbf{76.49 \pm 0.25}$	$\mathbf{85.44 \pm 0.20}$	93.01 ± 0.05	94.49 ± 0.00	$\mathbf{75.82 \pm 0.00}$	$\mathbf{92.74 \pm 0.00}$	$\mathbf{84.24 \pm 0.05}$	**86.32**

Table 2: Feature extraction performance (Accuracy) of MICRO-Graph compared with other self-supervised learning (SSL) baselines on molecule property prediction benchmarks. Use pre-trained models to extract graph representations for each data and train linear classifiers on top of them.

4.1.3 Model Configuration and Hyperparameters

We use the Deeper Graph Convolutional Networks (DeeperGCNs) proposed in Li et al. (2020), the state-of-the-art GNN architectures, as our base GNN encoder. For fair comparison, we use the same model hyperparameters (number of layers is 5 and number of hidden dimension is 300) recommended in Li et al. (2020) for our method and all baselines. We pre-train our model using Adam optimizer for 100 epochs, with batch size (number of graphs per batch) as 512. During fine-tuning, we train the model for 100 epochs with batch size as 32, and select the output model with highest validation performance.

4.2 Evaluation Result

The evaluation results under transfer learning setting and feature extraction setting is illustrated in Table 1 and Table 2. For both setting, the average performance of our proposed *MICRO-Graph* is higher than all the self-supervised learning baselines, and it achieves the highest results on most datasets. For transfer learning setting, we can gain about 2.8% performance enhancement against non-pretrain baseline. This shows the effectiveness of our self-supervised learning framework for pre-training GNNs, especially when the label data is scarce.

4.3 Ablation Study

In the following, we conduct a series of ablation studies to systematically analyze how the motif learning can benefit the contrastive learning.

4.3.1 Whether Motif is helpful for subgraph sampling?

As previously discussed, the main difference of our contrastive framework with existing works is that we leverage the subgraph-to-graph view as contrastive prediction task. As there are multiple ways to generate subgraphs, we thus study whether our proposed motif-guided subgraph segmenter can indeed help contrastive learning.

We implement three widely adopted heuristic subgraph sampling baseline: random walk (RW) and K-hop sampling (K-hop). We replace our Motif-guided Subgraph Segmenter (MSS) component with these two heuristic sampling algorithm, and conduct experiments under the Transfer Learning setting. All the other settings are the same.

As is shown in Table 3, replacing our Motif-guided Subgraph Segmenter (MSS) with heurstic subgraph sampler will significantly influence the performance. With random walk or k-hop sampling, subgraph-to-graph contrastive learning can only bring in 0.3-0.5% average performance enhance-

Sampler	bace	bbbp	clintox	hiv	sider	tox21	toxcast	Average
RW	73.61 ± 2.53	82.24 ± 1.99	75.63 ± 2.86	73.06 ± 1.29	55.88 ± 1.69	76.14 ± 0.56	63.44 ± 0.76	71.42 (+0.3%)
K-hop	73.24 ± 2.65	82.65 ± 1.78	76.76 ± 3.88	73.48 ± 1.41	55.67 ± 1.51	76.01 ± 0.69	63.34 ± 0.94	71.59 (+0.5%)
MSS	$\mathbf{76.16 \pm 2.51}$	$\mathbf{83.78 \pm 1.77}$	$\mathbf{77.50 \pm 3.35}$	$\mathbf{75.51 \pm 0.67}$	$\mathbf{57.28 \pm 1.09}$	$\mathbf{76.68 \pm 0.36}$	$\mathbf{65.42 \pm 0.62}$	**73.19 (+2.8%)**

Table 3: Ablation study: analyzing the influence of subgraph sampler.

# Motif	bace	bbbp	clintox	hiv	sider	tox21	toxcast	Average
5	75.81 ± 2.38	82.65 ± 1.97	76.95 ± 2.44	74.53 ± 1.12	56.78 ± 1.64	$\mathbf{77.01 \pm 0.89}$	64.45 ± 0.59	72.59 (+2.0%)
20	$\mathbf{76.16 \pm 2.51}$	$\mathbf{83.78 \pm 1.77}$	$\mathbf{77.50 \pm 3.35}$	$\mathbf{75.51 \pm 0.67}$	57.28 ± 1.09	76.68 ± 0.36	$\mathbf{65.42 \pm 0.62}$	**73.19 (+2.8%)**
100	75.98 ± 2.36	83.68 ± 1.62	76.91 ± 2.57	75.06 ± 0.92	$\mathbf{57.32 \pm 1.12}$	76.98 ± 0.55	65.39 ± 0.69	73.04 (+2.6%)

Table 4: Ablation study: analyzing the influence of different motif numbers.

ment against non-pretrain, which is far less than the one with MSS. This shows that the key of our overall performance enhancement of is not only based on utilizing the subgraph-to-graph view, but also highly depends on the informative subgraphs generated with the help of motifs. For the details of sample strategies and the subgraphs sampled by each strategies, please refer to Appendix B.

4.3.2 Whether Motif number will influence contrastive Learning?

Number of motifs is an important hyperparameter in our motif mining framework. We thus conduct ablation study with three different number of motifs, 5, 20, and 100. As is illustrated in Table 4, with different number of motifs, our MICRO-Graph framework can consistently enhance the transfer performance by a large margin. Among the three numbers used, the middle one (20) gives the best result on average.

Figure 3: Top-6 frequently occurred motifs, represented by their closest subgraph.

4.4 Visualization of the learned Motifs

We further shows some learned motifs by collecting the closest subgraph to them. As is illustrated in Figure 3, the *MICRO-Graph* automatically learns some motifs that are similar to important function groups in molecule domain, such as Benzene ring and Acetate. This shows that our framework can indeed learn reasonable and meaningful motifs. For a complete list of all the learned motifs, please refer to Appendix A.

5 Conclusion

In this paper, we propose *MICRO-Graph* to pre-train GNN in a self-supervised manner that can automatically extract graph motifs from large-scale graph datasets. In addition, the learned motifs can guide to generate more informative subgraphs, and help conduct graph-to-subgraph contrastive learning. The motif learning and contrastive learning are mutually reinforced, and eventually help pre-train a more generalizable GNN encoder. By pre-training on ogbn-molhiv molecule dataset with our proposed *MICRO-Graph*, we can learn meaningful motifs that fit existing molecular functional groups. Meanwhile, the pre-trained GNN can be used to fine-tune on seven chemical property prediction benchmarks, yielding 2.8% average improvement over non-pretrained GNNs and outperforming other self-supervised pre-training baselines.

REFERENCES

Yunsheng Bai, Hao Ding, Yang Qiao, Agustin Marinovic, Ken Gu, Ting Chen, Yizhou Sun, and Wei Wang. Unsupervised inductive graph-level representation learning via graph-graph proximity, 2019.

Jin Chen, Wynne Hsu, Mong Li Lee, and See-Kiong Ng. Nemofinder: Dissecting genome-wide protein-protein interactions with meso-scale network motifs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 106–115, 2006.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1231–1239, 2012.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=HJlWWJSFDH.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020c.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77, 2020. URL https://transacl.org/ojs/index.php/tacl/article/view/1853.

Nadav Kashtan, Shalev Itzkovitz, Ron Milo, and Uri Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns, 2020.

Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

Nicolò Navarin, Dinh V. Tran, and Alessandro Sperduti. Pre-training graph neural networks with kernels, 2018.

Carl O Pabo, Ezra Peisach, and Robert A Grant. Design and selection of novel cys2his2 zinc finger proteins. *Annual review of biochemistry*, 70(1):313–340, 2001.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Graph contrastive coding for graph neural network pre-training. In *KDD*, 2020.

Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.

Falk Schreiber and Henning Schwöbbermeyer. Frequency concepts and pattern detection for the analysis of motifs in networks. In *Transactions on computational systems biology III*, pp. 89–104. Springer, 2005.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization, 2019.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax, 2018.

Lichen Wang, Bo Zong, Qianqian Ma, Wei Cheng, Jingchao Ni, Wenchao Yu, Yanchi Liu, Dongjin Song, Haifeng Chen, and Yun Fu. Inductive and unsupervised representation learning on graph structured objects. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkem91rtDB.

Sebastian Wernicke. Efficient detection of network motifs. *IEEE/ACM transactions on computational biology and bioinformatics*, 3(4):347–359, 2006.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A TOP K CLOSESET SUBGRAPHS TO LEARNED MOTIFS

Examples of the first 10 learned motifs of the ogbg-molhiv dataset is shown in Figure 4 and 5.

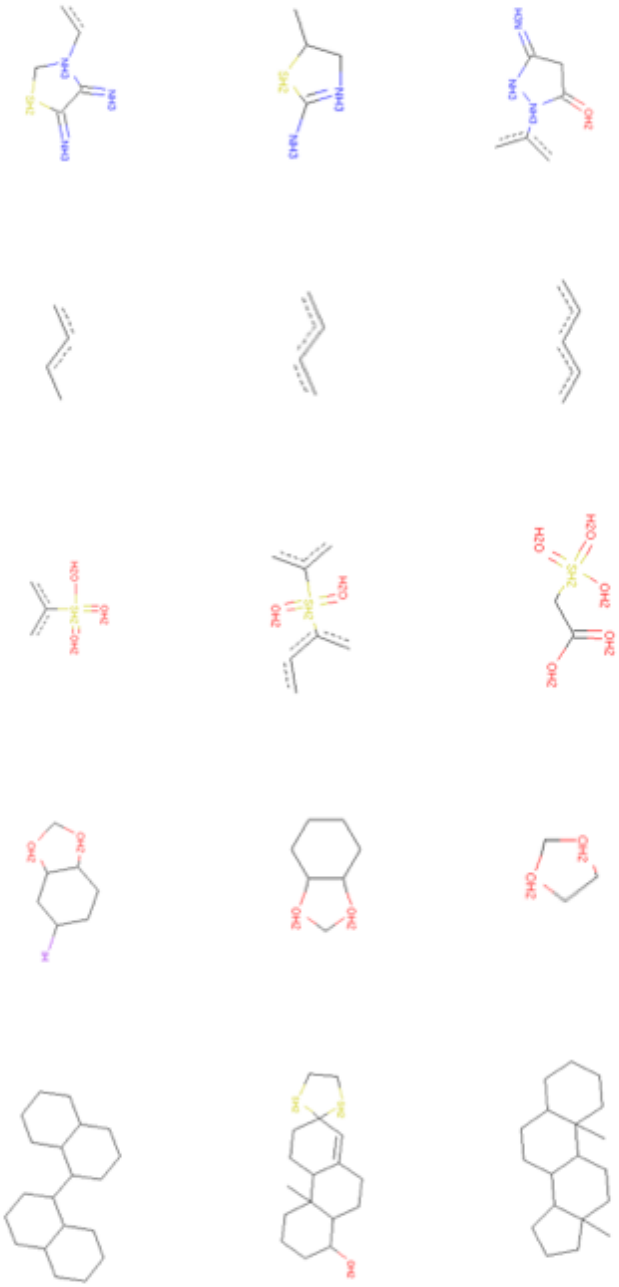

Figure 4: Motif 1-5, represented by top k closest subgraphs to the learned motif representations. Each row represents a motif, represented by some subgraphs that is closest to these motifs. Three columns indicates top 1, top 2, and top 3 most similar subgraph respectively.

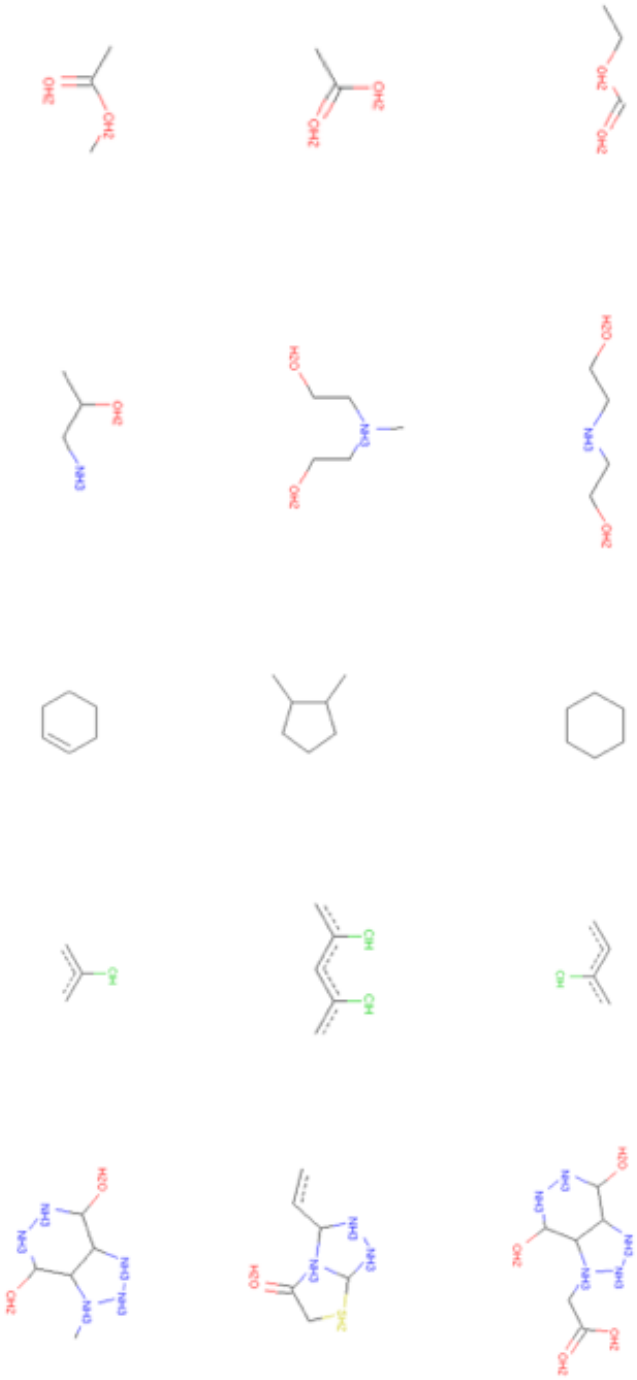

Figure 5: Motif 6-10, represented by top k closest subgraphs to the learned motif representations. Each row represents a motif, represented by some subgraphs that is closest to these motifs. Three columns indicates top 1, top 2, and top 3 most similar subgraph respectively.

B SAMPLING STRATEGIES AND SAMPLED SUBGRAPHS

Here we describe the details of our heuristic sampling strategies.

For random walk, we use a random walk length uniform in [10, 40]. Starting from a randomly selected seed node, we randomly select its neighborhood as next hop, until reaching the walk length threshold.

For K-hop neighbors, we pick hop number k to be 1 or 2 with equal probability. Starting randomly selected seed node, we collect all the neighbors within k hop as the sampled subgraph.

We also shows some subgraph examples generated by these two heuristic strategies and our proposed motif-guided subgraph segmenter in Figure 6. From the sampled subgraphs, we can see that random walk is more likely to generate chains, while k-hop sampling is more likely to generate half part of a Benzene ring. Neither of these two heuristic approaches can successfully generate a complete and clean functional group, and the generated subgraphs are not that meaningful. On the contrary, our motif-guided sampler can succssfully generate a complete benzene ring and two other molecule substructures. This intuitively explains why the graph-to-subgraph contrastive learning can only work with our proposed subgraph segmenter.

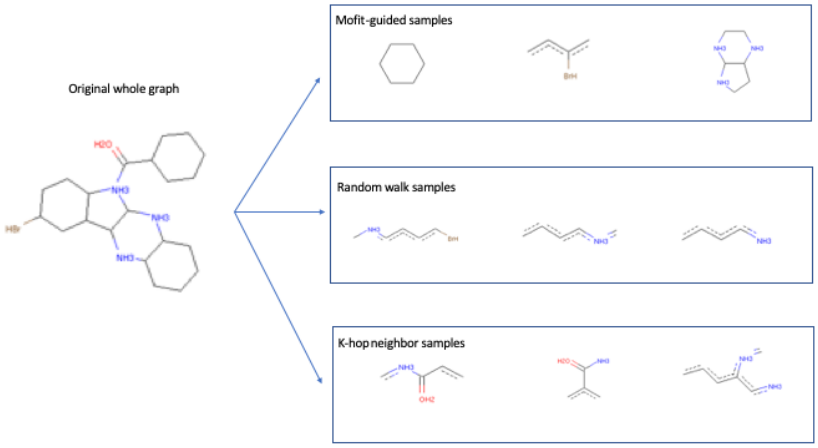

Figure 6: Comparison between different sampling strategies. The original graph is shown on the left. Samples produced by three different sampling strategies are shown on the right. The top row shows the samples by our motif-guided segmenter. The following two rows corresponding to random walk samples and k-hop samples.

C CHEMICAL PROPERTY PREDICTION BENCHMARKS

In our experiments, we evaluated model performance on seven Open Graph Benchmark (OGB) molecule property prediction datasets. We provide a synopsis of each downstream task dataset from Hu et al. (2020b) below:

- **bace:** Qualitative binding results for a set of inhibitors of human β-secretase 1.
- **bbbp:** Blood-brain barrier penetration (membrane permeability).
- **clintox:** Qualitative data classifying drugs approved by the FDA and those that have failed clinical trials for toxicity reasons.
- **hiv:** Experimentally measured abilities to inhibit HIV replication.
- **sider:** Database of marketed drugs and adverse drug reactions (ADR), grouped into 27 system organ classes.

Dataset	bace	bbbp	clintox	hiv	sider	tox21	toxcast
# graphs	1513	2039	1477	41127	1427	7831	8576
# nodes	51577	49068	38637	1049163	48006	145459	161088
# edges	111536	105842	82372	2259376	100912	302190	161088
# tasks	1	1	2	1	27	12	617

Table 5: Statistics on number of graphs, nodes, edges, and tasks in each OGB molecule dataset.

- **tox21:** Toxicity data on 12 biological targets, including nuclear receptors and stress response pathways.
- **toxcast:** Toxicology measurements based on over 600 in vitro high-throughput screenings.

Table 5 summarizes important statistics of the OGB molecule datasets related to the number of graphs, the size of graphs, and number of properties that require prediction for each molecule. For further information on the OGB datasets (e.g. node and edge features), please refer to Hu et al. (2020b) and Hu et al. (2020a).

D A SYNTHETIC DATASET TO STUDY MOTIF LEARNING

In addition to studying the pre-training in chemical domain, we also construct a synthetic dataset to that align with our assumptions to verify the effectiveness of the propose method. We assume there exist K graph motifs, and each whole graph can be representated by certain combinations of these motifs. Following this rule, we first select some graph structures as motifs, and randomly sample some combinations, and generate graphs (as is illustrated in Figure 7) by combining the motifs, with some randomly added or deleted nodes and edges. Each graph will also be assigned a corresponding one-hot vector label, whose dimension is the total number of combinations we generated for the dataset. Eight subgraph templates and examples of generated whole graphs are shown below. As is illustrated in Figure 8, our *MICRO-Graph* can successfully learn the underlying graph motifs and templates without any annotations.

The advantage of constructing such synthetic dataset is that we can know the underlying ground-truth of graph motifs and combination rules. We believe on top of this toy dataset, more complex syntax and semantic knowledge can be incorporated for automatic motif and even rule mining communities.

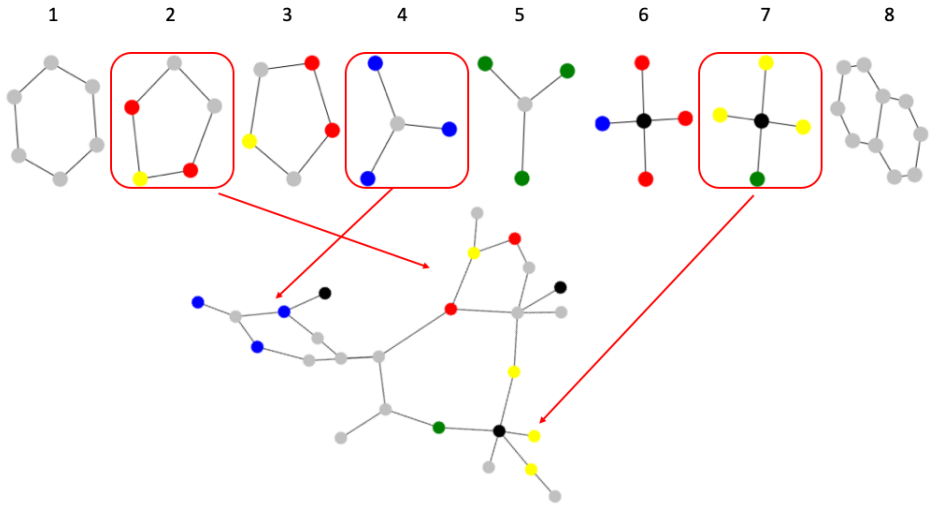

Figure 7: Example of a synthetic graph. The upper 8 graphs are the base motifs, and we generate this graph with a combination [2, 4, 7]. Different colors represent different node features.

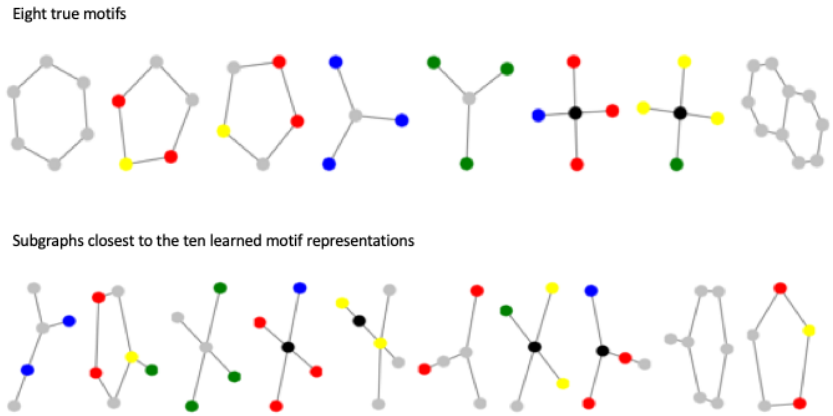

Figure 8: Comparison of the eight motifs used to generate the synthetic dataset and subgraphs closest to the ten learned motif representations