Structure-aware Self-supervised Graph Representation Learning

Lingwen Liu[1], Peng Cao ✉[1,2,3], Guangqi Wen[1], Zhuolin Jia[1], Jinzhu Yang[1,2,3], Weiping Li[4], and Osmar R. Zaiane[5]

[1] Computer Science and Engineering, Northeastern University, Shenyang, China
{2201839,2110658,2472012}@stu.neu.edu.cn
[2] Key Laboratory of intelligent Computing in Medical Image of Ministry of Education, Northeastern University, Shenyang, China
[3] National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Shenyang 110819, China
{caopeng,yangjinzhu}@cse.neu.edu.cn
[4] Software and Microelectronics, Peking University
wpli@ss.pku.edu.cn
[5] Alberta Machine Intelligence Institute, University of Alberta
zaiane@cs.ualberta.ca

Abstract. Graph-level tasks such as classification and regression are critical in various domains, including citation network analysis, protein molecular analysis, and social network analysis, where understanding complex network structures is important. Although Graph Self-Supervised Learning (GSL) has proven effective for these tasks, it often falls short in capturing subtle yet crucial structural information. This paper presents GraphTEL, a novel Self-Supervised Graph Topology Embedding Learning framework designed to overcome the limitations of GSL by enhancing sensitivity to structural nuances through explicit learning of graph topological patterns. GraphTEL consists of dual topology learning, which explores both global and local topological characteristics, alongside well-designed pretext tasks optimized by topology-aware loss function. Experimental results and visualizations show that GraphTEL produces robust and discriminative graph-level embeddings, outperforming existing methods in graph-level tasks and offering a powerful tool for analyzing complex connectivity patterns across diverse applications. Significant improvements are observed in graph-level tasks. Our model outperforms the AD-GCL/JOAO by an average increase of 7.84%/7.25% and 0.098/0.122 on graph classification/regression tasks, respectively, demonstrating its advantages in comparison to the state-of-the-art models. The code is available at https://github.com/IntelliDAL/Graph/tree/main/GraphTEL.

Keywords: Graph Structure Learning; Self-supervised Learning; Graph Classification; Graph Regression

1 Introduction

Graph self-supervised learning (GSL) is a machine learning paradigm where a graph model learns from the inherent structure of graph data without explicit labeling. It is crucial for learning robust and generalizable embeddings that can be applied across diverse applications, including brain network classification [20], protein molecular analysis[15], and social network analysis[14].

Understanding and exploiting the graph structure in GSL is essential for accurately analyzing complex graphs. Unfortunately, existing GSL methods face significant challenges in capturing and leveraging graph structural information effectively. For example, when analyzing molecular structures (Fig. 1(a)), distinguishing between different molecular categories based on topological patterns is critical. Most mask-based GSL approaches like GraphMAE[6] use Message Passing Graph Embedding Encoders (e.g. GCN[8] and GAT[19]) to generate embeddings, which are then used to predict and reconstruct masked subgraphs, with the aim of training the encoders to produce generalizable embeddings. However, they fail to recognize key topological differences (Fig. 1(b)), where the model incorrectly embeds the molecule, ignoring the topological patterns. Due to the complex characteristics and non-Euclidean properties of graph data, it is non-trivial to directly transfer the self-supervised learning paradigms designed for CV/NLP to graph data analytics. Unlike grid-like or sequential data, graph data necessitates encoders and pretext tasks that are specifically designed to capture its unique topological properties. The core question is: **"What essential features should be exploited via graph self-supervised learning for better facilitating various graph downstream analysis tasks?"** In this paper, we want to study and rethink the process of graph self-supervised learning. Our empirical study shows that it is critical to accurately capture the graph topological information. As demonstrated in Fig. 1(c), our model accurately provides the node embeddings that can reflect the crucial topological differences.

To sufficiently capture the representation of graph topology in GSL, the following questions come to mind: (1) how to make the heterogeneous graphs into a consistent scale? (2) How can the graph topology be captured exactly? To address these challenges in a self-supervised manner and provide robust topological representations for various downstream tasks, we developed the self-supervised Graph Topology Embedding Learning framework (GraphTEL). Concretely, GraphTEL mainly consists of: ***(i)*** **Structure alignment** for transforming structurally heterogeneous graphs into a consistent scale via a learnable clustering method. ***(ii)*** **Dual topology learning** for explicitly encoding global and local topological features. The global topology reveals macro-structural features of the graph, such as community topological modes crucial for downstream tasks, whereas local topology captures the micro-structural nuances around individual nodes, essential for understanding node roles and predicting relationships between nodes. It includes a global topology learning submodule, which uses edge convolution operation to learn connectivity patterns and their implications on the global structure, and a local topology learning submodule, which identifies subtle structural differences and preserves the locally topological detail via motifs. ***(iii)***

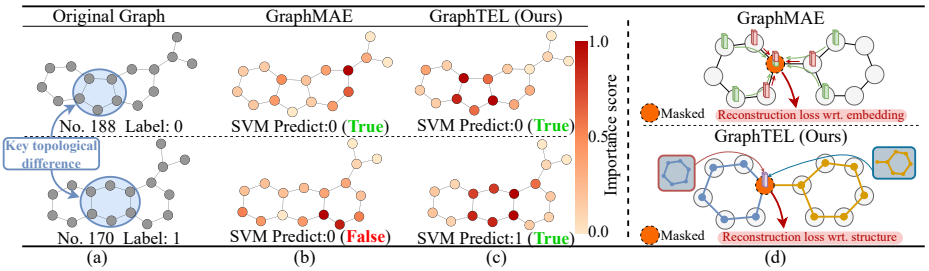


Fig. 1: Intuitive visualization of graph representations and prediction results generated by GraphMAE[6] and our GraphTEL. The importance scores of nodes are produced by GNNExplainer[26]. Deeper node colors represent higher importance scores. The graph representations learned by the pre-trained GraphMAE[6] and GraphTEL are fed into the SVM classifier to predict the label. (a) The two original graphs with different labels can be identified at the topological level by recognizing pentagonal and hexagonal topological patterns, indicated by blue circles. (b) GraphMAE[6] captures node features but fails to discover the key topological differences, leading to incorrect predictions by SVM. (c) Our model GraphTEL, focusing on learning graph topology, successfully captures the key topological differences, enabling SVM to correctly labeling the two graphs. (d) The paradigms of node information aggregation VS. graph structure learning.

Design of topology-aware loss function for better guiding the topology learning. To enable robust graph representation learning, we propose two masking strategies, i.e. edge random masking and rule-based random walk masking for perturbing the topology at the local and global levels, respectively. Additionally, we incorporate contrastive learning into the reconstruction task, to encourage the model to focus on key information and discriminative topology for reconstructing the masked graph. We summarize our main contributions as follows:

1. We pinpoint the key challenge in existing GSL methods: the lack of effective topological consideration. To address this, we propose a novel Self-supervised Graph Topology Embedding Learning framework (GraphTEL) to create powerful topological representations, which is crucial for understanding the complex connectivity patterns in graphs.

2. To preserve and explore local topology in graphs, we develop a motif mapping and motif convolution process to improve the understanding of local topological patterns and alleviate the issue of critical information loss occurring in the graph alignment.

3. The results demonstrate the advantages of the proposed method in graph classification and regression tasks. Moreover, the visual analysis highlights the powerful capabilities of highlighting significant topological patterns within the graph, providing insights that illustrate why GraphTEL works. Such capability is of scientific merit to the community of graph self-supervised learning.

2 Related Works

2.1 Graph Structure Learning

Graph structure learning (GSL) has been investigated for message passing graph neural networks (MPGNNs) with the aim to boost their performance on various downstream tasks by sufficiently exploring graph topological information[22,32]. The usual paradigm for most GSL works is to use MPGNNs to model graph adjacency matrix via node embedding and optimize them under the supervision of downstream classification tasks. However, MPGNNs are highly dependent on node features and over-emphasize node proximity, resulting in the inability to fully reflect graph topology[1,31]. These limitations restrict the performance of existing GSL methods.

2.2 Graph Self-supervised Learning

To address the issue of insufficient supervision, self-supervised methods for graph embedding learning have been developed, primarily using pretext tasks. These methods fall into two main categories: graph masked autoencoders and graph contrastive learning. Graph masked autoencoders, such as GraphMAE[6], focus on learning representations by reconstructing masked node features from visible parts[30]. While effective, these methods often neglect the graph's topological information[29,4], focusing primarily on node embeddings via propagation. Conversely, graph contrastive learning methods aim to maximize mutual information using the graph structure. GraphCL[28] learns unsupervised representations through graph augmentations such as node dropping and edge perturbation. GraphCLA[13] advances this by generating multiple augmentation views to mitigate the heavy reliance on manual augmentations. However, these methods generally fail to explicitly capture graph structures[15,7], which limits their ability to learn effective representations from graphs.

3 Method

Given a graph $G = \{V, E\}$ where V is the node set and E is the edge set, GraphTEL aims to learn a topology embedding mapping $H_t = f(E)$, subsequently utilized in graph-level tasks $Y = g(V, H_t)$. As shown in Fig. 2, GraphTEL consists of a topology learning network (TLN) and self-supervised pretext tasks.

3.1 Topology Learning Netowrk (TLN)

Structure Alignment Heterogeneous graphs exhibit varying structural semantics due to differences in scale, *e.g.*, larger graphs tend to have more complex structural information than smaller ones. To address this inconsistency, we introduce a structure alignment operation, which aligns heterogeneous graphs into scale-unified graphs with consistent node numbers while preserving crucial global

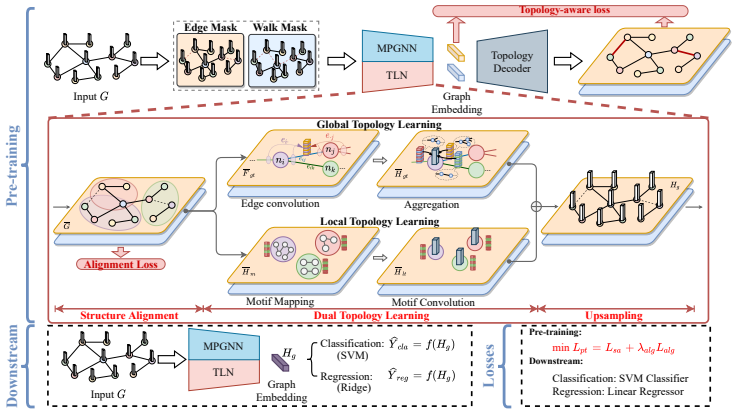


Fig. 2: GraphTEL consists of 1) Topology learning network (TLN) for aligning inconsistent graphs, exploring the global and local topological characteristics and generating topological embedding; 2) Self-supervised pretext tasks, consisting of edge and walking masking for disturbing graphs, topology decoder for structure reconstruction, and topology-aware loss for guiding topology embedding learning.

topology. Specifically, we utilize a learnable cluster assignment matrix $S \in \mathbb{R}^{N \times P}$, where P represents the size of the alignment structure. We determined the optimum of P based on the performance of the validation dataset. The learning of the cluster assignment matrix can be formalized as $S = \Phi(A_{\cdot}^{v}, X)$, where X is the initial node features and $\Phi(\cdot)$ is stacked GraphSAGE layers with an activation function **Softmax**. Then, the adjacency matrix $\bar{A} \in \{0,1\}^{P \times P}$ of the aligned graph can be obtained via $\bar{A} = S^{\mathsf{T}} A S$. Note that the optimization for learning S is non-convex, thereby we introduce two regularizations. Each node is expected to be assigned to one cluster, we regularize the entropy of S by $L_H = -\frac{1}{n}\sum_{i=1}^{n} H(S_i)$, where S_i is the i-th row of S. In addition, an L_2 regularization is added to avoid extreme assignment imbalances, described by $L_E = \|\frac{1}{n}\sum_{i=1}^{n} S_i\|_2$. Thus the regularization $L_{alg} = L_H + L_E$.

Dual Topology Learning The topological information can reflect the inherent properties of graphs. However, MPGNN-based graph learning methods fall short of exploring sufficient topological information, and therefore necessitate an effective solution to capture various topological modes. We argue that there exists global and local topology in a graph. Specifically, global topology indicates the macro connectivity modes of the whole graph, while local topology represents the micro connectivity modes between a node and its neighbors. Thus, to fully exploit the graph topology, we propose a dual topology learning module, which consists of structure alignment, global/local topology learning, and an upsampling operation.

Global Topology Learning. We consider that there exist two global topological associations: an explicit association involving e_{pq} connecting the node p and node q, and an implicit association within edge sets $e_{p\cdot}$ and $e_{\cdot q}$. We therefore

design a specialized edge convolution operation that leverages graph global topology information by considering explicit and implicit associations, which is an edge aggregation operation with multiple cross-shaped filters for graph spatial locality. Each cross-shaped filter consists of a $1 \times P$ horizontal filter and a $P \times 1$ vertical filter. Then, the two filters are individually element-wise multiplied and then added to obtain a feature map. Formally, let $M^{(l)}$ be the feature matrix obtained by the l-th layer of edge convolution, and $M^{(0)} = \bar{A}$. The edge convolution layer can be formulated as:

$$M^{(l)} = \Psi_H(M^{(l-1)}, w_H^{(l-1)}, C_e) + \Psi_V(M^{(l-1)}, w_V^{(l-1)}, C_e) \tag{1}$$

where $w_H^{(l-1)}$ and $w_V^{(l-1)}$ denote the learnable weights of $1 \times P$ horizontal filter and $P \times 1$ vertical filter in the $(l-1)$-th layer. C_e represents the channel number, and we set $C_e = h$. Thus, we obtain the edge embedding $\bar{F}_{gt} \in R^{P \times P \times h}$ learned by edge convolution. The operations consider both explicit associations in edge connections and implicit associations within node neighborhoods, enhancing the learning of global topological associations in graphs.

Finally, we generate the global topology embeddings $\bar{H}_{gt} \in R^{P \times h}$ via an aggregation layer $\bar{H}_{gt} = (\bar{F}_{gt}, w_U^{(l)}, C_{gt})$, where $w_U^{(l)}$ denotes the learnable weights of $1 \times P$ filter and $C_{gt} = h$.

Local Topology Learning. We also consider learning local topology information within aligned supernodes. However, our structure alignment method indicates global connections but loses local topology details. To preserve and leverage the local topological information, we utilize undirected graph motifs to represent each supernode's topology as a unique vector. Traditional motif collection methods rely on simplistic statistical counting, struggling with complex topological associations. Additionally, increasing vertex numbers brings more structural variations in motifs (2 motif types for 3-vertex, and 6 motif types for 4-vertex[9]), necessitates a unified high-dimensional representation method for motifs of varying vertex sizes.

To this end, we devise a specific motif mapping approach. We first identify motifs via DotMotif[12] and generate motif Laplician matrix set with K_m elements for each supernode: $\mathcal{T} = \{\nabla_3, \nabla_4, ..., \nabla_k\}$. Each $\nabla_k = \{\nabla_k^1, \nabla_k^2, ..., \nabla_k^{n_k}\}$ signifies the motif Laplician matrix set for the discovered n_k motifs corresponding to k-vertex and $\nabla_k^i = I - (\widetilde{D}_k^i)^{-\frac{1}{2}} A_k^i (\widetilde{D}_k^i)^{-\frac{1}{2}}$, where $A_k^i \in [0,1]^{k \times k}$ is the motif adjacency matrix for k-vertex and $(\widetilde{D}_k^i)^{-\frac{1}{2}}$ is the diagonal degree matrix of A_k^i. In this work, we set the maximum value of k to 6, thus $K_m = 4$. We then apply multiple filters to the motif Laplician matrices $\mathcal{T}$, formulated as:

$$H_m^p = \prod_{j=3}^{k} \mathbf{Concat}\left(\frac{1}{n_j} \sum_{i=1}^{n_j} w_j \nabla_j^i\right) \tag{2}$$

where $w_j \in R^{j \times j \times C_m}$ represents the learnable weights of the filter, and C_m is the channel number. This maps motifs into high-dimensional vectors, resulting in $H_m^p \in R^{K_m \times C_m}$, which represents the topology of the p-th supernode. The motif

representation $\bar{H}_m \in R^{P \times K_m \times C_m}$ is obtained by concatenating $H_m^1, H_m^2, ..., H_m^P$. Subsequently, we design a motif convolution for better capturing the local topology. Let $M_{lt}^{(l)}$ be the feature map of the l-th layer and $M_{lt}^{(0)} = H_m$, the motif convolution is given by: $M_{lt}^{(l)} = \Psi_{lt}(M_{lt}^{(l-1)}, w_{lt}^{(l-1)}, C_{lt})$, where $w_{lt}^{(l)}$ denotes the learnable weights of $1 \times P$ filter and $C_m = h$. Thus, we obtain the local topology embedding $\bar{H}_{lt} \in R^{P \times h}$ from the last layer output. Finally, the topology embedding $\bar{H}_t$ of graph $\bar{G}$ is generated by $\bar{H}_t = \bar{H}_{gt} + \bar{H}_{lt}$.

Upsampling To obtain the embedding aligned with the scale of original graphs, we propose an upsampling layer to map $\bar{H}_t \in R^{P \times h}$ into $H_t \in R^{N \times h}$. Similar to the structure alignment operation, the upsampling operation learns an upsampling matrix $\bar{S} \in R^{N \times P}$, therefore the topological representation H_t of the original graph G can be obtained by $H_t = \bar{S}\bar{H}_t$.

Node features are also essential for identifying graph structural properties. For example, hydrophobicity is a molecular structural property associated with the node properties[3,18]. To utilize structural features more accurately, we suggest incorporating node features. Thus, the final node embedding $F = \mathbf{Concat}(H_t, H_n)$ can be obtained, where the node embedding H_n is obtained from stacked graph convolution layers. Finally, the graph embedding $H_g \in R^h$ is generated by $H_g = \sum_{i=1}^{N} \Psi_G(F_i, w_g, C_g)$, where w_g denotes the learnable weights of $1 \times 2h$ filter, and the channel number $C_g = h$.

3.2 Self-supervised learning pretext tasks

Graph Masking Scheme. Graph masking fosters robust representation learning and highlights essential information[10,17]. We implement two strategies: edge random masking for local topology and rule-based random walk masking for global dependencies, both aimed at enhancing representation learning via graph topology. For the Edge Random Masking, we randomly drop a subset of edges to generate a masked graph based on a mask ratio of λ_r, resulting in visible graph $A_r^{\mathbf{v}}$ and masked graph $A_r^{\mathbf{m}}$. For Rule-Based Random Walk Masking, we initiate random walks , with each walk spanning $0.5\lambda_w \times |E|$, $|E|$ is the total number of edges, and λ_w is the masking ratio. To ensure robust masking without overlaps, we enforce three rules: each path must have non-repeating nodes, maintain a minimum node degree, and avoid intersections between paths. From generated paths, we select two, forming $E_w^{\mathbf{m}}$ and $E_w^{\mathbf{v}}$. This approach forces the model to detect vital structural patterns for predicting missing connections.

Topology Decoder. The topology decoder is proposed to reconstruct the masked edges via embeddings learned from visible graph structures. With the learned node embedding F, we use stacked GCN layers, in which the last layer output is latent embedding $\hat{X}$, then the edge reconstruction can be defined as $\hat{A} = \hat{X}\hat{X}^T$.

Topology-aware loss. We propose a novel topology-aware loss to ensure that the learned graph embeddings retain the original graph topology while maintaining distinguishability between different graphs. Firstly, we define: Reconstruction loss $L_{rec} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (1 - sim(A_i^{\mathbf{m}}, \hat{A}_i^{\mathbf{m}}))$, $sim(\cdot, \cdot)$ denotes the scaled cosine error (SCE),

inspired by GraphMAE[6]. $A_i^{\mathbf{m}}$ and $\hat{A}_i^{\mathbf{m}}$ represent the masked edge parts of the i-th original and reconstructed graph, respectively, with $A^{\mathbf{m}} = A - A^{\mathbf{v}}$ and $\hat{A}^{\mathbf{m}} = \hat{A} - \hat{A}^{\mathbf{v}}$. Contrastive loss $L_{ct} = L_{nc}^{ct} + L_{gc}^{ct} = \sum_{g,g' \in G} L_{con}(F_g^r, F_g^w, F_{g'}) + L_{con}(H_g^r, H_g^w, H_{g'})$ is used to capture structural information from two perturbed graphs with different masking schemes, L_{con} is the NCE loss. The node contrastive loss L_{nc}^{ct} focuses on intra-graph learning, treating embeddings of the same nodes from different masked graph versions as positive pairs, and different nodes as negative pairs. The graph contrastive loss L_{gc}^{ct} focuses on inter-graph learning, where embeddings of the same graph are treated as positive pairs, while embeddings of different graphs are treated as negative pairs. Then, we assume:
1) When graphs are successfully reconstructed, with L_{rec} nears 0, we enhance the discriminability of the high-quality topological embeddings via L_{ct}.
2) When graphs fail to be reconstructed, with L_{rec} nears 1, we focus more on improving the quality of topological embeddings, while less focus on L_{ct}.
Thus, the topology-aware loss L_{sa} is defined as follows:

$$L_{sa} = L_{rec} + (1 - L_{rec}) \cdot L_{ct} \tag{3}$$

This objective function jointly considers structural property modeling and the discriminative capacity of the embeddings through a combination of reconstruction and contrastive loss. The total loss for the pre-training task is given by $L_{pt} = L_{sa} + \lambda_{alg} L_{alg}$, where L_{alg} represents the alignment loss, and λ_{alg} is the weight for the alignment loss.

4 Experiments and Results

4.1 Experiment Settings

Datasets. For classification, we use all available datasets from[6]. The datasets chosen have diversity in density and degree. The detailed information is shown in Table 1. For regression, we choose datasets from[15]. This also ensures that we did not choose only the datasets on which our method performs better.
Comparative Models. We select supervised method GAT[19] and self-supervised methods (GraphMAE[6], GraphCL[28], MVGCL[5], JOAO[27], AutoGCL[25], AD-GCL[15], CI-GCL[16], GraphCLA[13], LAMP[2], SEGA[21] and DGPM[24]). It is worth noting that we perform pre-training for all the comparable GSL methods, then freeze the weights and optimize the head part during the downstream tasks, i.e. SVM for classification following the settings of GraphMAE[6], or Ridge for regression following the settings of AD-GCL[15]. This allows for better observation of model behavior and ensures a fair comparison with baseline models, avoiding the potential negative effects of fine-tuning.
Hyperparameter settings. The model training uses the Adam optimizer with a batch size of 32. The random mask ratio and walk mask ratio are both set to 0.3. The learning rate is 0.001. The maximum number of training epochs is 200 for pre-training and 100 for linear probing.

Table 1: The graph properties of different datasets.

Attribute	IMDB-B	IMDB-M	PRO	CO	MUT	REDDIT-B	NCI1
Number	1000	1500	1113	5000	188	2000	4110
Node	19.8K	19.5K	43.5K	372.5K	97.9K	859.2K	122.3K
Edge	386.1K	395.6K	162.1K	49.1M	202.5K	4M	265.5K
Density	1.98E-03	2.08E-03	1.72E-04	7.08E-04	4.22E-05	1.08E-05	3.55E-05
Max degree	540	352	50	2K	20	12.2K	8
Min degree	4	4	2	4	2	4	2
Avg degree	39	40	7	263	4	9	4

Table 2: Experiment results of graph classification *wrt.* accuracy (%), where the colors of **red** and blue denote the best results and the second best results, respectively. $D = \frac{2|E|}{|V|(|V|-1)}$ denotes the graph density. $\Delta = \max_{v \in V} \deg(v)$ represents the maximum degree. Graph property details can be seen in Table 1.

Methods	IMDB-B $D = 1.98e^{-3}$ $\Delta = 540$	IMDB-M $D = 2.08e^{-3}$ $\Delta = 352$	PRO $D = 1.72e^{-4}$ $\Delta = 50$	CO $D = 7.08e^{-4}$ $\Delta = 2K$	MUT $D = 4.22e^{-5}$ $\Delta = 20$	REDDIT-B $D = 1.08e^{-5}$ $\Delta = 12.2K$	NCI1 $D = 3.55e^{-5}$ $\Delta = 8$
GAT[19]	70.50±2.30	47.80±3.10	——	67.40±2.90	89.40±6.10	85.20±3.30	——
GraphMAE[6]	75.52±0.66	51.63±0.52	75.30±0.39	80.32±0.46	88.19±1.26	88.01±0.19	**80.40±0.30**
GraphCL[28]	71.14±0.44	48.58±0.67	74.39±0.45	71.36±1.15	86.80±1.34	89.53±0.84	77.87±0.41
JOAO[27]	70.83±0.25	——	74.55±0.41	69.50±0.36	87.67±0.79	86.42±1.45	78.07±0.47
AutoGCL[25]	73.30±0.40	——	75.80±0.36	70.12±0.68	88.64±1.08	88.58±1.49	——
AD-GCL[15]	72.33±0.56	49.89±0.66	73.81±0.46	73.32±0.61	89.70±1.03	85.52±0.79	69.67±0.51
CI-GCL[16]	73.50±0.89	50.23±0.23	74.90±0.77	74.50±0.09	87.36±1.31	88.80±0.82	79.09±1.11
GraphCLA[13]	72.33±0.44	49.53±0.25	74.48±0.25	71.73±0.28	89.33±0.85	88.69±0.72	72.97±0.58
LAMP[2]	75.39±0.52	45.36±0.63	75.99±0.95	76.53±0.77	90.01±0.89	89.13+0.32	78.28±0.56
SEGA[21]	73.58±0.26	50.82±0.46	74.50±0.32	74.12±0.45	90.20±1.10	88.73+0.24	77.21±0.75
DGPM[24]	75.77±0.54	50.93±0.51	75.05±0.61	79.28±0.44	90.60±0.98	88.07+0.85	80.15±0.67
GraphTEL	**77.40±0.31**	**54.45±0.26**	**77.10±0.41**	**82.12±1.56**	**91.91±0.68**	**91.74±0.33**	80.33±0.54

4.2 Results

(a) Graph-level Analysis Tasks. o evaluate our model's effectiveness in graph classification and regression, we compare it with state-of-the-art methods. Table 2 shows that GraphTEL consistently outperforms existing models. With a focus on topology embedding, GraphTEL surpasses GraphMAE by an average of 2.63% and GraphCL by 5.49% across six datasets, highlighting the importance of topological information in graph embedding learning. Both GraphMAE and GraphTEL, as generative-based GSL methods, outperform contrastive-learning-based GSL models, demonstrating the advantage of generative self-supervised pre-training. Notably, GraphTEL's topology reconstruction further enhances graph understanding and embedding quality. Furthermore, two compelling properties of GraphTEL are observed:

i) The explicit topology learning mechanism of GraphTEL offers a powerful complement to MPGNNs, particularly on datasets with high density and complex topology. GraphTEL shows the best performance among the algorithms for almost all datasets/metrics, except for NCI1 dataset, which exhibits sparsely topological characteristics. Specifically, after examining the graph properties of density (D) and maximum degree (Δ) shown in Tables 1&2,

Table 3: Experiment results of graph regression *wrt.* RMSE, where **red** and blue denote the best and the second-best results, respectively.

Methods	molesol	mollipo	molfreesolv
GIN[23]	1.71±0.18	1.08±0.02	7.53±2.12
GraphCL[28]	1.27±0.09	0.91±0.02	7.68±2.75
MVGCL[5]	1.43±0.15	0.96±0.04	9.02±1.98
JOAO[27]	1.28±0.12	0.86±0.03	5.13±0.72
NAD-GCL[15]	1.39±0.06	0.95±0.02	5.84±0.87
AD-GCL[15]	1.21±0.08	0.84±0.02	5.15±0.62
GraphTEL	**1.14±0.07**	**0.81±0.04**	**4.97±0.72**

Table 4: Ablation studies on the IMDB-B dataset. GT: global topology learning, LT: local topology learning.

Methods	Topology		Node embedding	Loss			ACC(%)
	GT	LT	H_n	L_{rec}	L_{nc}^{ct}	L_{gc}^{ct}	
Naive GCN			√				70.04
GraphGL	√	√					71.00
GraphGLR -w/o masking	√	√		√			73.66
GraphGLR	√	√		√			75.11
GraphGLNR	√	√	√	√			75.84
GraphTEL -w/o L_{gc}^{ct}	√	√	√	√	√		76.18
GraphTEL -w/o L_{nc}^{ct}	√	√	√	√		√	76.95
GraphTEL -w/o LT	√		√	√	√	√	76.61
GraphTEL -w/o GT		√	√	√	√	√	75.25
GraphTEL	√	√	√	√	√	√	**77.40**

we note that NCI1 has a large number of nodes and edges, yet the maximum degree is relatively low (8) and the network density is also low (3.55e-5). This suggests that the connections between nodes in NCI1 graphs are not dense, and the topological patterns are sparse, hence topological features do not provide sufficient knowledge. Although the value of D in the REDDIT-B dataset is also low, its maximum degree ($\Delta = 12.2K$) significantly exceeds the other datasets, indicating the presence of highly connected central nodes and the topology is complicated, modeling topology information benefits the embedding learning.

ii) GraphTEL sufficiently utilizes graph information in conditions of informational scarcity. Although the MUTAG dataset has low D and Δ, it is a relatively smaller dataset with only 188 samples. Fully leveraging its topological information facilitates better graph-level embedding learning. The COLLAB dataset, on the other hand, has a large number of graph samples but lower numbers of nodes and edges per graph (see Table 1). Learning generalizable features from smaller graphs requires sufficiently exploiting the potential graph structure, demonstrating that GraphTEL can effectively leverage the essential information from graphs to learn generalizable representations.

We also evaluate the performance of our model on the graph property regression task on the *molesol*, *mollipo*, and *molfreesolv* datasets, shown in Table 3. Notably, for the *molfreesolv* dataset, our model achieves the best result with an RMSE of 4.97, significantly improving upon the second-best result 5.15. These results highlight the effectiveness of GraphTEL in handling both graph classification and regression tasks across different datasets, corroborating its versatility and robustness. The compelling experiments also confirm our initial hypothesis about the importance of modeling global and local topological information in graph representation learning.

(b) Ablation Study. To demonstrate the effectiveness of each component, we perform extensive ablation experiments. The results are reported in Table 4. First of all, all the GSL models outperform the baselines without pre-training, *i.e.*, naive GCN and GraphGL. This demonstrates the effectiveness of the proposed components in our model. Moreover, it can be observed that the performance drops as each component or loss is removed, demonstrating the necessity of these components and losses. The results further reveal that the masking strategy is

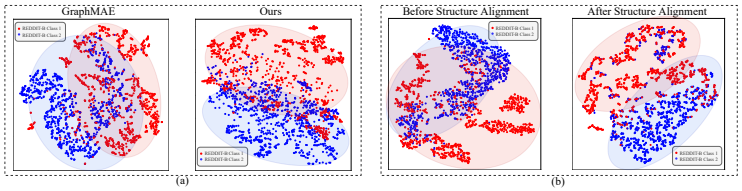


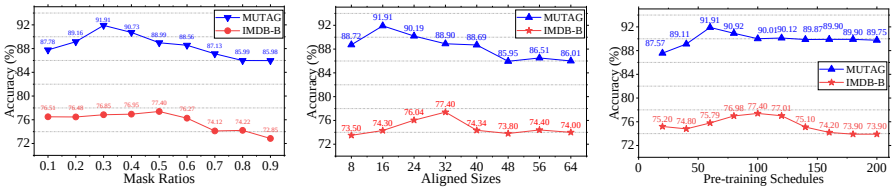
Fig. 3: Visualization using REDDIT-B, each color represents a class. (a) Graph embeddings. (b) Class distribution before/after structure alignment.

crucial in graph reconstruction, as its absence leads to a significant performance drop (GraphGLR -w/o masking). Incorporating node features (GraphGLNR) improves performance over using only topological information (GraphGLR), suggesting that the graph node features help identify graph topological properties, thereby improving topological information exploration. Integrating contrastive learning (GraphTEL -w/o L_{gc}^{ct} or GraphTEL -w/o L_{nc}^{ct}) enhances the model's ability to learn discriminative information, beneficial for classification tasks. Both local and global topology learning components are useful for improved performance (GraphTEL -w/o LT or GraphTEL -w/o GT), confirming the necessity of exploring graph topology from local and global views. In addition, our local topology learning essentially preserves the local graph topology lost in the process of coarsening the graph, which leads to a substantial improvement in performance (76.61%→77.40%).

(c) Analysis of Graph Embeddings. To further demonstrate the effectiveness of our model, we provide a t-SNE[11] visualization in Fig. 3(a) showing that the embeddings learned by our model provide more discriminative power. This suggests that: (i) Topological learning enables the model to focus on crucial topological patterns and obtain distinguishable representations. (ii) Capturing topological information explicitly offers deeper insights into structurally semantic information, and avoids relying heavily on node features.

(d) The Influence of Structure Alignment. Although previous results demonstrate the effectiveness of GraphTEL, we are still interested in the qualitative analysis of structure alignment. As illustrated in Fig. 3(b), we can see the distribution visualization of the original and aligned graph representation using t-SNE[11], respectively. Compared with the distribution before the structure alignment, the points of different labels can clearly be separated from each other through the proposed structure alignment, which validates that the inconsistent graph structures containing varying hierarchies of structure information negatively affect the structure learning, and our structure alignment enables the graph representation more discriminative.

(e) Hyperparameter Sensitivity. We conduct a hyperparameter sensitivity analysis to systematically investigate how the factors affect the performance, which can be seen in Fig. 4. ***i.)*** **Mask ratios.** Fig. 4(a) shows the influence of mask ratios varying from 0.1 to 0.9 for the classification performance on the IMDB-B and the MUTAG datasets. Our model performance generally improves



(a) (b) (c)

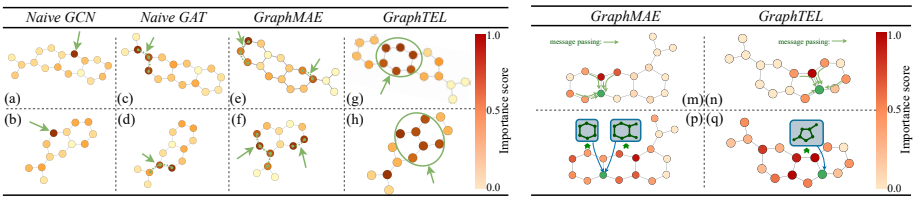
Fig. 4: The variation of classification performance with different masking ratios (a), sizes of aligned structures (b) and pre-training epochs (c).

as the mask ratio increases, but it starts to degrade when the mask ratio becomes excessively large. Interestingly, we find it differs from other mask-based self-supervised methods[6,10], which maintain high performance even with high mask ratios. The reason lies in the fact that other methods focus on learning crucial information through aggregating node features through MPGNNs. A higher mask ratio aids in removing information redundancies, such as unnecessary node features or edges that don't contribute to message passing. However, our model focuses on exploring graph topological patterns, thereby higher mask ratios lead to disregarding crucial topological patterns. ***ii.)*** **The sizes of aligned structures.** We study the impact of the aligned structure size on the IMDB-B dataset and MUTUG dataset. Fig. 4(b) shows the classification results with different aligned structure sizes. It can be found that the optimal aligned structure size for IMDB-B dataset is 32, while the optimal size of the MUT dataset is 16, from which we can conclude that the optimal alignment structure size P is related to the size of the graph. The larger the graph, the larger the alignment size required. ***iii.)*** **Influence of pre-training schedules.** We perform a pre-training schedule analysis in Fig. 4(c) to discuss how it affects the classification performance. It can be seen that performance improvement is achieved when the pre-training schedules increase. Nevertheless, as the pre-training schedules further increase, the reconstruction task rapidly achieves convergence and tends to be overfitting, which decreases the downstream classification performance. It demonstrates that a suitable pre-training schedule benefits graph embedding learning.

(f) Model complexity. We also show the comparative results of computational efficiency. Naive GCN has 10.52K parameters, an inference time of 1.01 seconds, and consumes 1.21 GB of GPU memory. GraphMAE, with 12.79K parameters, shows a slightly longer inference time of 1.04 seconds and uses 1.59 GB of memory. GraphTEL, containing 11.97K parameters, achieves an inference time of 1.03 seconds and requires 1.43 GB of GPU memory. It demonstrate that GraphTEL achieves better performance while maintaining model efficiency.

4.3 Analysis of Salient Nodes

To answer the question of *what information our GraphTEL captures*, we derive node importance scores via the GNNExplainer[26] from the DGL library. Initially,

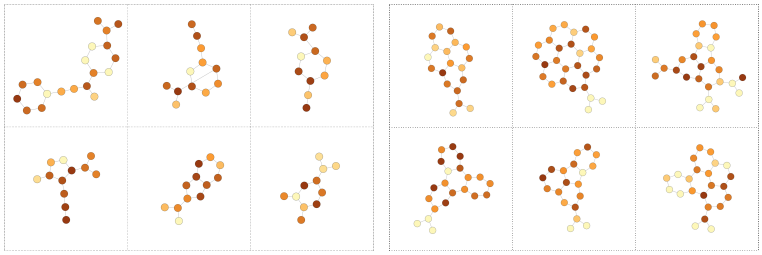


(a) Node importance analysis (b) Neighbor importance analysis

Fig. 5: Importance visualization of salient nodes across different models (Naive GCN, Naive GAT, GraphMAE, and GraphTEL) on the MUTAG dataset.

we feed the graphs into a GNNExplainer initialized by a pre-trained model, which outputs edge scores. It is assumed that a high edge score indicates the neighboring node connected by the edge is of great importance to the specific node; conversely, if all associated edges of a node have high importance scores, it suggests that the node itself is highly significant. We perform salient node importance analysis and neighbor importance analysis, respectively. We selected the MUTAG dataset due to its suitable graph scale and distinct topological features.

(a) Importance Analysis of Salient Node. To show the ability of each model to capture underlying graph information, we perform a graph-level salient node importance analysis by aggregating the importance scores of all connecting edges associated with each node to determine its importance score. As shown in Fig. 5(a)-(d), the Naive MPGNNs like GCN and GAT, condense graph information into nodes via message passing, relying on only a few salient nodes to capture the graph's essence. However, their simplicity limits their ability to identify complex topological patterns crucial for certain graph tasks. GraphMAE improves node importance distribution across the graph, as shown in Fig. 5(e)(f), but it lacks explicit topological guidance[6], which limits its effectiveness in tasks requiring complex structure understanding. In contrast, GraphTEL demonstrates superior capability in identifying and emphasizing key topological patterns. On the MUTAG dataset (Fig. 5(g)(h)), GraphTEL captures distinct connection patterns among different topologies that are crucial for predicting graph types. To further illustrate GraphTEL's ability to differentiate between graph categories, we analyze node importance distribution under two categories (label=0 and label=1) in Fig. 6. Category 1 (Label=0) typically consists of simpler, more linear graphs, and GraphTEL highlights linear or branched features, with important nodes sequentially reflecting these characteristics. Conversely, Category 2 (Label=1) contains more circular and complex patterns, where key nodes often occupy central or connecting positions, crucial for graph connectivity. GraphTEL adeptly identifies these circular or clustered topological features, including complex arrangements like hexagons. This visualization underscores that GraphTEL not only focuses on independent topological modes but also captures the correlation between graph components. It is especially critical for datasets like MUTAG, where topological patterns are closely linked to mutagenic properties[3].



(a) Salient node importance (Label=0)

(b) Salient node importance (Label=1)

Fig. 6: Visualization of salient node importance for two classes on MUTAG.

(b) Importance Analysis of Salient Node for Feature Aggregation. Fig. 5(m)-(q) illustrates critical differences in how graph information is utilized between GraphMAE and GraphTEL. GraphMAE aggregates features from neighbors and often fails to capture the nuanced but topologically significant patterns. As shown in Fig. 5(m) and (n), importance scores are distributed across neighbor nodes of the salient node, which dilutes the potential topological role of these nodes. Unlike MPGNNs, GraphTEL employs a topological pattern aggregation strategy. In GraphTEL, the salient node's feature update is significantly influenced by a subset of topologically related neighbor nodes, which form a series of topological patterns. Thus, GraphTEL can identify nodes based on their topological significance within the graph, as reflected in Fig. 5(p)(q), where high-scoring nodes demonstrate discernible topological patterns. By effectively recognizing and leveraging crucial topology, GraphTEL addresses the core limitation in MPGNNs through an explicit topology discovery approach. Such an approach not only aids in achieving higher performance in tasks requiring advanced topological awareness but also enhances the model's interpretability.

5 Conclusion

We design a novel graph self-supervised graph topology embedding learning framework named GraphTEL for graph-level tasks. The proposed GraphTEL focuses on exploring graph topological information from both global and local perspectives to generate topology embedding by well-designed dual topology learning and pretext tasks (masked graph reconstruction, contrastive learning and topology-aware loss), to extract potentially discriminative information for graph-level tasks. Extensive experiments demonstrate the effectiveness of our model in graph-level tasks, highlighting its advantages over state-of-the-art approaches. We also conduct various analysis especially node importance visualization to prove the effectiveness of our model. Our findings highlight the necessity of graph topology representation learning in graph-level tasks, offering a robust self-supervised graph representation learning model.

6 Acknowledgments.

This research was supported by the National Natural Science Foundation of China (No.62076059), the Science and Technology Joint Project of Liaoning province (2023JH2/101700367) and the Fundamental Research Funds for the Central Universities (No. N2424010-7).

References

1. Ai, B., Qin, Z., Shen, W., Li, Y.: Structure enhanced graph neural networks for link prediction. arXiv preprint arXiv:2201.05293 (2022)
2. Chen, X., Li, S., Wu, J.: Uncovering capabilities of model pruning in graph contrastive learning. In: ACM Multimedia 2024 (2024)
3. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry **34**(2), 786–797 (1991)
4. Feng, J., Wang, Z., Li, Y., Ding, B., Wei, Z., Xu, H.: Mgmae: Molecular representation learning by reconstructing heterogeneous graphs with a high mask ratio. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 509–519 (2022)
5. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: International conference on machine learning. pp. 4116–4126 (2020)
6. Hou, Z., Liu, X., Cen, Y., Dong, Y., Yang, H., Wang, C., Tang, J.: Graphmae: Self-supervised masked graph autoencoders. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 594–604 (2022)
7. Kim, D., Baek, J., Hwang, S.J.: Graph self-supervised learning with accurate discrepancy learning. In: 36th Conference on Neural Information Processing Systems, NeurIPS 2022. pp. 1–19 (2022)
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations. pp. 1–14 (2017)
9. Li, C., Wei, W., Feng, X., Liu, J.: Research of motif-based similarity for link prediction problem. IEEE Access **9**, 66636–66645 (2021)
10. Li, J., Wu, R., Sun, W., Chen, L., Tian, S., Zhu, L., Meng, C., Zheng, Z., Wang, W.: What's behind the mask: Understanding masked graph modeling for graph autoencoders. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1268–1279 (2023)
11. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
12. Matelsky, J.K., Reilly, E.P., Johnson, E.C., Stiso, J., Bassett, D.S., Wester, B.A., Gray-Roncal, W.: Dotmotif: an open-source tool for connectome subgraph isomorphism search and graph queries. Scientific Reports **11**(1), 13045 (2021)
13. Pu, X., Zhang, K., Shu, H., Coatrieux, J.L., Kong, Y.: Graph contrastive learning with learnable graph augmentation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
14. Salha-Galvan, G., Lutzeyer, J.F., Dasoulas, G., Hennequin, R., Vazirgiannis, M.: Modularity-aware graph autoencoders for joint community detection and link prediction. Neural Networks **153**, 474–495 (2022)

15. Suresh, S., Li, P., Hao, C., Neville, J.: Adversarial graph augmentation to improve graph contrastive learning. Advances in Neural Information Processing Systems **34**, 15920–15933 (2021)
16. Tan, S., Li, D., Jiang, R., Zhang, Y., Okumura, M.: Community-invariant graph contrastive learning. In: Forty-first International Conference on Machine Learning
17. Thakoor, S., Tallec, C., Azar, M.G., Munos, R., Veličković, P., Valko, M.: Bootstrapped representation learning on graphs. In: ICLR 2021 Workshop on Geometrical and Topological Representation Learning (2021)
18. Tian, Y., Wang, X., Yao, X., Liu, H., Yang, Y.: Predicting molecular properties based on the interpretable graph neural network with multistep focus mechanism. Briefings in Bioinformatics **24**(1), bbac534 (12 2022)
19. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
20. Wen, G., Cao, P., Bao, H., Yang, W., Zaiane, O.: Mvs-gcn: A prior brain structure learning-guided multi-view graph convolution network for autism spectrum disorder diagnosis. Computers in Biology and Medicine **142**, 105239 (2022)
21. Wu, J., Chen, X., Shi, B., Li, S., Xu, K.: Sega: Structural entropy guided anchor view for graph contrastive learning. In: International Conference on Machine Learning. pp. 37293–37312. PMLR (2023)
22. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems **32**(1), 4–24 (2020)
23. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
24. Yan, P., Song, K., Jiang, Z., Kang, Y., Lin, T., Sun, C., Liu, X.: Empowering dual-level graph self-supervised pretraining with motif discovery. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 9223–9231 (2024)
25. Yin, Y., Wang, Q., Huang, S., Xiong, H., Zhang, X.: Autogcl: Automated graph contrastive learning via learnable view generators. Proceedings of the AAAI Conference on Artificial Intelligence **36**(8), 8892–8900 (Jun 2022)
26. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
27. You, Y., Chen, T., Shen, Y., Wang, Z.: Graph contrastive learning automated. In: International Conference on Machine Learning. pp. 12121–12132 (2021)
28. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 5812–5823. Curran Associates, Inc. (2020)
29. Yu, W., Huang, M., Wu, S., Zhang, Y.: Ensembled masked graph autoencoders for link anomaly detection in a road network considering spatiotemporal features. Information Sciences **622**, 456–475 (2023)
30. Zhang, Q., Wang, Y., Wang, Y.: How mask matters: Towards theoretical understandings of masked autoencoders. In: Advances in Neural Information Processing Systems. vol. 35, pp. 27127–27139 (2022)
31. Zhong, Z., Li, C.T., Pang, J.: Hierarchical message-passing graph neural networks. Data Mining and Knowledge Discovery **37**(1), 381–408 (2023)
32. Zhu, Y., Xu, W., Zhang, J., Liu, Q., Wu, S., Wang, L.: Deep graph structure learning for robust representations: A survey. arXiv preprint arXiv:2103.03036 **14** (2021)