Transitivity-Preserving Graph Representation Learning for Bridging Local Connectivity and Role-Based Similarity

Van Thuy Hoang, O-Joun Lee*

Department of Artificial Intelligence, The Catholic University of Korea
{hoangvanthuy90, ojlee}@catholic.ac.kr

Abstract

Graph representation learning (GRL) methods, such as graph neural networks and graph transformer models, have been successfully used to analyze graph-structured data, mainly focusing on node classification and link prediction tasks. However, the existing studies mostly only consider local connectivity while ignoring long-range connectivity and the roles of nodes. In this paper, we propose Unified Graph Transformer Networks (UGT) that effectively integrate local and global structural information into fixed-length vector representations. First, UGT learns local structure by identifying the local substructures and aggregating features of the k-hop neighborhoods of each node. Second, we construct virtual edges, bridging distant nodes with structural similarity to capture the long-range dependencies. Third, UGT learns unified representations through self-attention, encoding structural distance and p-step transition probability between node pairs. Furthermore, we propose a self-supervised learning task that effectively learns transition probability to fuse local and global structural features, which could then be transferred to other downstream tasks. Experimental results on real-world benchmark datasets over various downstream tasks showed that UGT significantly outperformed baselines that consist of state-of-the-art models. In addition, UGT reaches the third-order Weisfeiler-Lehman power to distinguish non-isomorphic graph pairs.

Introduction

Graph neural networks (GNNs) and graph transformers have effectively solved graph analysis tasks, such as node classification and link prediction (Kipf and Welling 2017; Dwivedi and Bresson 2021). GNNs learn representations by iteratively aggregating features of neighbouring nodes with uniform or attentional weights (Kipf and Welling 2017; Brody, Alon, and Yahav 2022; Hoang et al. 2023a). Unlike GNNs, graph transformer models learn node representations using self-attention, encoding pair-wise connections between nodes, and have shown better performance than variants of GNNs (Dwivedi and Bresson 2021).

While the existing models have been applied in various domains, GNNs and graph transformers still have three inherent limitations. ***First***, graph transformer models lack the power to capture similarity between local structures. Some recent graph transformers (e.g., GT (Dwivedi and Bresson 2021) and SAN (Kreuzer et al. 2021)) apply graph theory-based structural features (e.g., Laplacian Eigenvectors) as positional encodings (PEs) to assist the model distinguish isomorphic substructures. However, although the PEs could assign distinguishable vector representations to substructures of individual nodes, they did not fully consider the structural similarity between the substructures. Node representations should be strongly correlated with their substructure similarity from the first iteration (layer) in order for the model to understand connectivity between neighbours and not just aggregate their feature vectors. In Figure 1, we assign the initial PEs to two nodes '1' and '3' with Laplacian Eigenvectors. Then, their vector representations would be treated as different representations even though they are structurally similar. This false signal can be a serious barrier to representation learning of graph structure and eventually reduce the performance of models. ***Second***, GNN aggregators and self-attention lack the power to learn global structural information. GNN variants, like GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), and GATv2 (Brody, Alon, and Yahav 2022), compose node embeddings based on only neighbours' features. Similarly, graph transformer models learn vector representations by encoding features of node pairs and substructures within k-hop distance using self-attention (Chen, O'Bray, and Borgwardt 2022). Although graph transformers have relatively larger aggre-

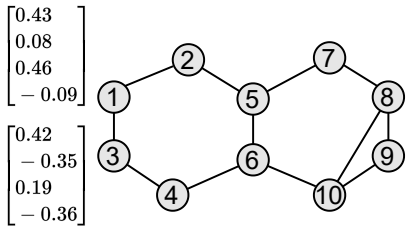


Figure 1: The Laplacian positional encoding cannot fully capture the similarity between the surrounding structure of adjacent nodes. Two adjacent nodes, '1' and '3', have different initial positional encodings even though they are structurally similar.

*Corresponding author: O-Joun Lee (Tel.: +82-2-2164-5516)

gation ranges in a layer than GNNs, neighbourhood aggregation has little ability to reveal long-range dependencies or role-based similarities between nodes (Lee and Jung 2020a; Jeon, Choi, and Lee 2022). ***Third***, the existing models lack the ability to integrate local and global structural features into a unified vector representation, which could then benefit various downstream tasks. Some GNNs and graph transformers, such as LSPE (Dwivedi et al. 2022) and GPS (Rampásek et al. 2022), try to capture higher-order substructures surrounding nodes within k-hop distance and consider them in messages or node features. However, global structural features can be extracted by using the roles of nodes (i.e., substructures rooted in nodes). Role-based similarity is not always synchronous with the similarity of node features, attributes, or classes. Also, nodes with similar roles do not mean that they are neighboring nodes, but rather the opposite (e.g., influential nodes in social networks) (Lee and Jung 2020b). Therefore, the different viewpoints of the different features create a significant barrier that poses challenging problems in integrating these features into a unified vector representation.

To overcome these limitations, we propose Unified Graph Transformer Networks (UGT), a novel graph transformer model that can represent both local and global structural features of graphs with a unified fixed-length vector. We first construct virtual edges, bridging the distant nodes with structural similarity. The main idea is to capture the long-range dependencies between distant nodes as long as they are structurally similar. To capture local structure, we propose structural identity to assist the model in addressing local non-isomorphic substructures and finding similar substructures. Third, we propose structural distance to estimate the role-based similarity between nodes. Accordingly, UGT can learn node representations such that nodes with structural similarity could be close in the latent space. Note that global structural information is measured by role-based distance, which could not combine with local structural information as it is considered based on k-hop distance. Therefore, to bridge the gap between local and global structural features, we utilise the p-step transition probability, which could reflect the global structural information and the distance between nodes as long as two nodes are connected. Our contributions are as follows:

- We propose Unified Graph Transformer Networks (UGT), which could learn both local and global structural information and fuse them into unified representations.
- We propose a sampling technique that constructs virtual edges and a structural distance measurement between distant nodes to capture global structural similarity.
- We propose a self-supervised learning task to bridge the conceptual gap between local and global structural features by preserving transition probabilities between nodes in multi-scales.
- We experimentally examined that UGT reaches the third-order Weisfeiler-Lehman (3d-WL) power to distinguish non-isomorphic graph pairs.

Related Work

This section discusses how existing models handle the above weaknesses compared to our UGT model. Several graph transformers attempt to find local structural similarities with neighbours by defining different PEs. For example, SAN uses learnable PEs instead of static Laplacian Eigenvectors combined with an attention mechanism to learn the local structures (Kreuzer et al. 2021). Graphformer adds node degree to node features and integrates edge features through Shortest-Path Distance attention bias (Ying et al. 2021). GPS uses some types of random walk positional encoding (RWPE) to find local similar substructures for each target node (Rampásek et al. 2022). SAT extracts k-hop sub-graph representations for each target node and utilises GNNs to update target node representations (Chen, O'Bray, and Borgwardt 2022). The existing models attempt to map different substructures into different representations or find local structural similarity from paths without considering the node roles and their structural distance (Hoang et al. 2023b). Unlike existing models, we propose structural identity (I) and structural distance (d), which could distinguish non-isomorphic substructures and find the structural similarity between nodes. The structural identity acts more effectively to find structurally similar neighbours since it can handle both local and global structures through the roles of nodes in a hierarchical manner.

Several studies have been proposed to capture the global structure information (Lee, Jeon, and Jung 2021). WRGAT aims to break the limitations of GNNs by using multi-relations between distant nodes in graphs (Suresh et al. 2021). GeoGCNs employ several embedding methods, such as Poincare and Struc2vec, and create suitable latent spaces, preserving the global structure information (Pei et al. 2020). While the GNN variants show the capability to capture global information, they mostly overlook sub-structure similarity. Several graph transformers aim to capture higher-order structures within k-hop distance, such as SAT (Chen, O'Bray, and Borgwardt 2022) and Graphformer (Ying et al. 2021). However, high-order structures are defined within k-hop distance, which ignores structural correlations between distant nodes. Unlike existing models, we construct virtual edges to connect distant nodes and learn their structural similarity. With an understanding of the local and global structure, UGT fuses them into fixed-size representations, which could address various downstream tasks. UGT is a new graph transformer framework that learns both local and global structural information and fuses them into fixed-length vector representations.

Unified Graph Transformer Networks

This section introduces a novel graph transformer model, namely UGT, which could learn local and global structures and fuse them into unified representations. We first introduce how to sample context nodes and then describe the design of UGT in detail. Lastly, we introduce self-supervised learning and fine-tuning tasks. The overall architecture of UGT is described in Figure 2.

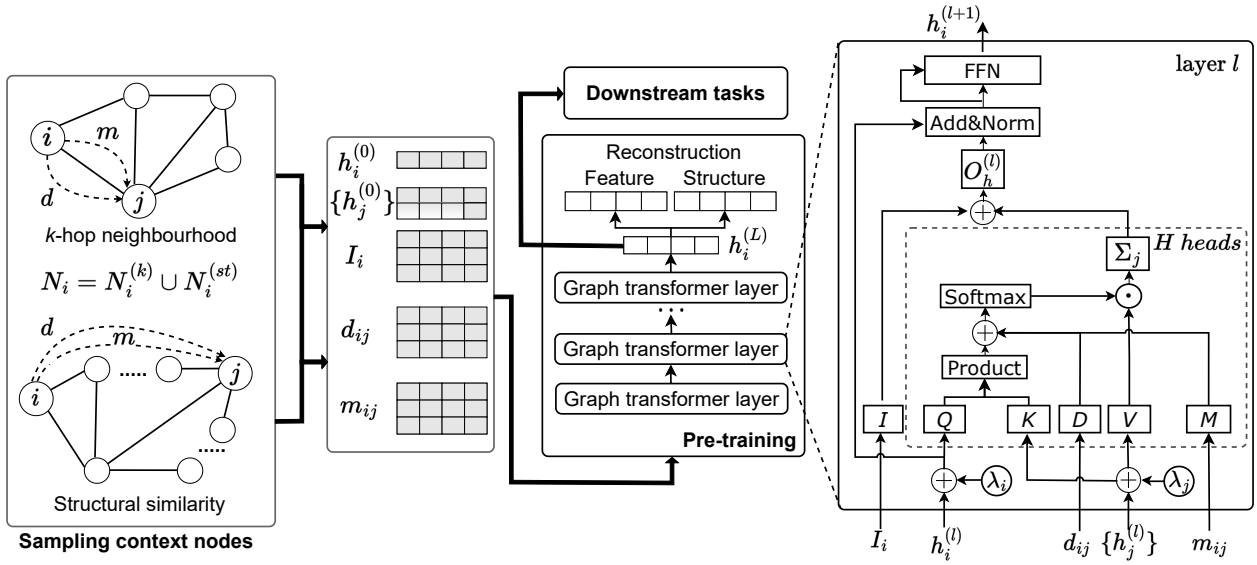


Figure 2: The overall architecture of UGT. UGT is composed of main blocks, including sampling context nodes, building modules I, d, and m, and pre-training blocks. The learned representations then could be used for various downstream tasks.

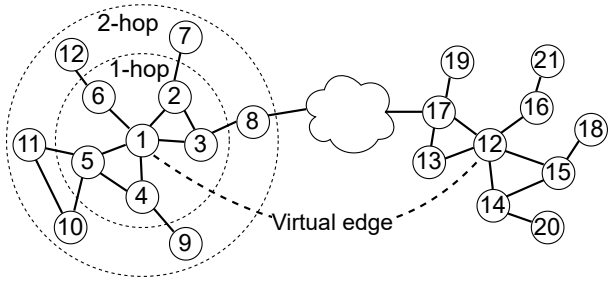


Figure 3: An example of UGT sampling strategy. We sample context nodes within k-hop neighbourhoods and virtual connections between distant nodes with structural similarity.

Sampling Context Nodes

Given a graph $G = (V, E)$ with node set V and edge set E, we aim to sample the context node v_j of each target node $v_i \in V$. The context node v_j could be sampled if the distance from v_i to v_j within k-hop, or both two nodes v_i and v_j are structurally similar. Formally, for each target node v_i, the set of neighbour of v_i can be defined as:

$$N_i \quad = \quad N_i^{(k)} \cup N_i^{(st)}, \tag{1}$$

where $N_i^{(k)}$ presents the neighbourhoods of v_i within k-hop distance, and $N_i^{(st)}$ refers to the set of nodes that are structurally similar to v_i. Figure 3 presents a simplified graph that contains a target node '1' and its context. For global sampling, we construct virtual edges between any two nodes if they are structurally similar. For example, nodes '1' and '12' are structurally similar with the same degree of five and connected to two triangles. Note that we only consider similarity based on the graph structure without using the node feature.

Let $I_k(v_i)$ denotes the set of a ordered degree sequence of v_i in 1 to k-hop. Note that $I_0(v_i)$ is the degree of v_i. By comparing the ordered degree sequences between two nodes v_i and v_j, we can impose a hierarchy to measure global structural similarity. Formally, the distant score between any two nodes v_i and v_j in a graph could be defined as:

$$s_k = f_k\left(I_k(v_i), I_k(v_j)\right), \tag{2}$$

$$S\left(v_i, v_j\right) = \exp\left(-\sqrt{s_k}\right), \tag{3}$$

where $f_k(\cdot, \cdot)$ is the structural distance between two sequences. Similar to Struc2vec (Ribeiro, Saverese, and Figueiredo 2017), we then use dynamic time warping (DTW) to measure the similarity between two ordered degree sequences. In most real-world networks, however, the degree distribution is highly asymmetric and shows a long tail distribution. To boost the connections between high-degree nodes, we introduce another score:

$$S\left(v_i, v_j\right) = \exp\left(-\sqrt{s_k}\right) + \exp\left(-\frac{1}{\sqrt{\left(d_{v_i} + d_{v_j}\right)}}\right), \tag{4}$$

where d_{v_i} presents the degree of node v_i. Accordingly, two nodes with higher scores are structural similarity, and if they have high degrees together, they are closer in the latent space.

Learning Unified Representations of Graphs

Input representation Given a graph G, the node feature $x_i \in R^{d_0 \times 1}$ of node v_i is mapped via a linear projection to d-dimensional hidden feature, as:

$$\hat{x}_i^0 \quad = \quad W_0 x_i + b_0, \tag{5}$$

where $W_0 \in R^{d \times d_0}$ and $b_0 \in R^d$ are the parameters of the linear projection, d_0 refers to the original feature of v_i.

Since we aim to make UGT can learn the global positional information context for each node in the whole graph, we add a linearly transformed positional embedding λ_i of dim k to node features, as:

$$\hat{\lambda}_i = W_1\lambda_i + b_1, \quad (6)$$

$$h_i^0 = \hat{x}_i^0 + \hat{\lambda}_i, \quad (7)$$

where $W_1 \in R^{d\times k}$ and $b_1 \in R^d$. Note that the positional embeddings are pre-calculated and added only for the first time, and we use Eigenvectors of the graph Laplacian matrix. Furthermore, we randomly flip λ during training to allow UGT to capture sign invariance.

Global structural self-attention We now propose a self-attention bias to encode local and global structural information. Since we aim to capture context nodes within a k-hop distance and distant nodes, the self-attention needs to understand the distance between node pairs in terms of k-hop distance and the structural distance. Formally, we define a self-attention between v_i and v_j for head k at layer l as:

$$\alpha_{ij}^{k,l} = \frac{\left(Q^{k,l}h_i^l\right)\cdot\left(K^{k,l}h_j^l\right)}{\sqrt{d_k}} + D_{ij}^{k,l} + M_{ij}^{k,l}, \quad (8)$$

where $Q^{k,l}$, $K^{k,l} \in R^{d_k\times d}$, $k = 1$ to H refers to the number of attention heads, h_i and h_j are the features of node v_i and v_j, respectively. $D_{ij}^{k,l}$ and $M_{ij}^{k,l} \in R^{d_k\times d}$ are linear transformed structural distance d_{ij} and transition probability m_{ij} between two nodes v_i and v_j, respectively. We now introduce strategies to construct d and m.

We define the structural identity of each target node based on hierarchical role-based information, including minimum, maximum, mean, and standard deviation degrees up to k-hop distance context of each node v_i. Formally, the structural identity of node v_i and the structural distance could be defined as:

$$I_i = \{\, d_{v_i}, T_1, T_2, \cdots, T_k\}, \quad (9)$$

$$d_{ij} = f\left(\left[\left\|I_i^{(q)} - I_j^{(q)}\right\|_2\right]^{-1}\right), \quad (10)$$

where d_{v_i} denotes the degree of node v_i, $T_k = \{\min_k, \max_k, \mu_k, \delta_k\}$ denotes the set of minimum, maximum, mean, and standard deviation values of the node degrees at k-th hop distance, $I_i^{(q)}$ refers to the q-th element of I_i, and f refers to a linearly transformed structural distance.

While structural distance can benefit the model by finding distant nodes with structural similarity, it cannot capture paths between the two nodes. To fill this gap, we present the transition probability-based distance m, which is the awareness of connectivity between any two nodes from a probabilistic perspective. In this way, the model could capture both the local and global structural information, leading to the power of UGT to capture any path between nodes. We define the transition probability distance m from v_i to v_j as:

$$m_{ij} = f\left(A_{ij}^1, A_{ij}^2, \cdots, A_{ij}^p\right), \quad (11)$$

where A_{ij}^p refers to the transition probability from v_i to v_j at p-step, and f refers to a linearly transformed transition probability.

Graph transformer layers The outputs of self-attention are then concatenated into one vector representation followed by a linear transformation. The node features of v_i are updated at layer l as:

$$\hat{h}_i^{l+1} = O_h^l \Big\|_{k=1}^{H}\left(\sum_{v_j\in N(v_i)} \tilde{\alpha}_{ij}^{k,l} V^{k,l} h_j^l\right), \quad (12)$$

where $\tilde{\alpha}_{ij}^{k,l} = \text{softmax}_j(\alpha_{ij}^{k,l})$, $Q^{k,l}$, $K^{k,l}$,$V^{k,l} \in R^{d_k\times d}$, $O_h^l \in R^{d\times d}$, and $\|$ refers to concatenation. The outputs then are passed to feed-forward networks (FFN) along with residual connections and layer normalization:

$$\hat{h}_i^{l+1} = \text{LN}\left(h_i^l + \hat{h}_i^{l+1}\right), \quad (13)$$

$$\hat{\hat{h}}_i^{l+1} = W_2^l \text{ReLU}\left(W_1^l \hat{h}_i^{l+1}\right), \quad (14)$$

$$h_i^{l+1} = \text{LN}\left(\hat{h}_i^{l+1} + \hat{\hat{h}}_i^{l+1}\right), \quad (15)$$

where $W_1^l \in R^{2d\times d}$ and $W_2^l \in R^{d\times 2d}$ are learnable parameters, and LN refers to layer normalization.

We introduce a role-based identity for each substructure that can help our UGT model be more expressive when distinguishing non-isomorphic substructures. It is worth noting that structural identity plays two roles: finding similar substructures and distinguishing isomorphic substructures. Formally, we add a structural identity to the self-attention output at each transformer layer, which could be defined as:

$$h_i^{l+1} = f\left(I_i^l\right) + O_h^l \Big\|_{k=1}^{H}\left(\sum_{v_j} \tilde{\alpha}_{ij}^{k,l} V^{k,l} h_j^l\right). \quad (16)$$

Self-supervised Learning Tasks

We now present self-supervised learning tasks that could train UGT on pretext tasks to extract graph structure without using any label information. Our objective is to learn representations that could capture both local and global structures between any nodes as long as they are structurally similar. As mentioned earlier, we aim to preserve relational structure information in p-step transition probability, combining local and global structures. The learned representations could then be used for solving different downstream tasks.

Given a connectivity between a target node v_i and a context node v_j, we aim to maximize the transition probability of paths connecting v_i and v_j against the probability of other pairs not from the graph. Similar to Grarep and Glove, we employ noise contrastive estimation, introduced by (Gutmann and Hyvärinen 2012). The loss function for transition probability matrix of v_i at p-step could be defined as:

$$L_1^p(v_i) = \left(\sum_{v_j} P_k\left(v_j|v_i\right)\log\sigma\left(z_i z_j\right)\right) + \lambda\mathbb{E}, \quad (17)$$

where $\mathbb{E} = E_{v_k\sim P_k(V)}\left[\log\sigma\left(-z_i z_k\right)\right]$, v_k refers to nodes obtained from negative sampling. We then assume the noise follows a uniform distribution, and this lead to a new transition matrix on a log scale that represents precisely the global

relation information between any two nodes in the graph at step p:

$$A'^{(p)} = \log\left(\frac{A^{(p)}_{i,j}}{\sum_t A^{(p)}_{t,j}}\right) - \log\left(\frac{|N_s|}{|V|}\right), \quad (18)$$

where $A'^{(p)}$ is the log scale probability matrix at step p that we aim to preserve, and N_s refers to the set of negative samples. Finally, the loss function for structural preservation at step p can be computed as:

$$L^p_1 = \left\| A^{'(p)} - Z^{(p)} \right\|^2_F, \quad (19)$$

where $Z^{(p)}$ is the score matrix that could be built by computing the cosine similarity between two vectors z_i and z_j.

Since node feature information could benefit downstream tasks, UGT should also learn the construction of node feature information. We define the node raw feature reconstruction-based loss term as:

$$L_2 = \frac{1}{|V|}\sum\nolimits_{v_i \in V} \|x_i - \hat{x}_i\|_2, \quad (20)$$

where x_i refers to the raw feature of node v_i, and $\hat{x_i} = FFN(z_i)$. We train the model in multi-task learning with the two losses, and the losses are linearly combined as:

$$L = \alpha \sum_p L^p_1 + \beta L_2, \quad (21)$$

where α and β are hyper-parameters.

Fine-tuning Tasks

The learned representations are then passed directly to solve downstream tasks. In this study, we present three downstream tasks, including node clustering, node classification, and graph-level classification. For the node clustering task, we used modularity as the loss function as with Tsitsulin et al. (2023). For the node classification, the representations of v_i from the transformer layers were passed to fully connected (FC) layers to get y_i as:

$$y_i = W_1 \text{ ReLU}\left(W_2 h^l_i\right), \quad (22)$$

where $W_1 \in R^{d \times C}$ and $W_2 \in R^{d \times d}$ are weight matrices, and C is the number of classes. In the graph classification, we employed average pooling to obtain graph features and likewise used FC layers to obtain the prediction output.

Complexity Analysis

The structural feature matrices (I and d) are pre-computed only at the first time. To construct the identity matrix I, we use the Breadth First Search (BFS) to find nodes within k-hop distance with the computational cost of $O(N + E)$. We use fast DTW to compute the distance between two degree sequences with the computational cost of $O(l)$, where l is the sequence length. We also reduce the computation by limiting neighborhood sizes in I, as real-world graphs are sparse. The cost of structural distance and the transition matrix is $O(N^2)$ in the worst case. For model steps, the computational cost for each layer in full dot-product attention is $O(N^2)$, similar to other graph transformer models.

Experiments

We conducted experiments to evaluate UGT versus GNN variants and graph transformer models for three tasks: node clustering, node classification, and graph-level classification. We also analyzed the power of our model by assessing UGT on isomorphism testing. We deliver an open-source implementation of UGT for the experiment reproductions[1].

Experimental Settings

Datasets For the node-level tasks, we used eleven publicly available datasets, which are homophily and heterophily graphs and grouped into three different domains, including Air-traffic networks (e.g., Brazil, Europe, and USA) (Ribeiro, Saverese, and Figueiredo 2017), Webpage networks (e.g., Chameleon, Squirrel, Actor, Cornell, Texas, and Wisconsin) (Pei et al. 2020), and Citation networks (e.g., Cora and Citeseer) (Sen et al. 2008). We used five publicly available datasets for the graph classification task, including Enzymes, Proteins, NCI1, NCI9 from TUDataset (Morris et al. 2020) and large-scale graph OGBG MolHIV (Hu et al. 2020). Furthermore, we used Graph8c and five Strongly Regular Graphs datasets (SRGs), which contain 1d-WL and 3d-WL equivalent graph pairs, respectively, for isomorphism testing (Balcilar et al. 2021).

Baselines The GNN variants included GCN (Kipf and Welling 2017), GCNII (Chen et al. 2020), GIN (Xu et al. 2019), GAT (Velickovic et al. 2018), GATv2 (Brody, Alon, and Yahav 2022), SAGE (Hamilton, Ying, and Leskovec 2017), Geom-S (Pei et al. 2020), WRGAT (Suresh et al. 2021), and DeeperGCN (Li et al. 2020). Furthermore, we compared UGT against recent transformers, including GT (Dwivedi and Bresson 2021), SAN (Kreuzer et al. 2021), SAT (Chen, O'Bray, and Borgwardt 2022), GPS (Rampásek et al. 2022), ANS-GT (Zhang et al. 2022), Graphormer (Ying et al. 2021), and GD-WL (Zhang et al. 2023). For isomorphism testing, we compare UGT against powerful models, i.e., ChebNet (Defferrard, Bresson, and Vandergheynst 2016), and GNNML3 (Balcilar et al. 2021).

Implementation Details We conducted each experiment ten times by randomly sampling training, validation, and testing sets of size 80%, 10%, and 10%, respectively. The results written in the tables were measured with means and standard deviation on the testing set over the ten cases. The experiments were done in two servers with four NVIDIA RTX A5000 GPUs (24GB RAM/GPU). The hyperparameters were tuned on the validation sets. For fair comparisons with the baselines, we conducted a search for the number of layers and the hidden dimension size with ranges of {2, 4, 8} and {32, 64, 128} with every model, respectively.

Performance Analysis

Evaluation on node clustering We first conducted an experiment on eleven benchmark datasets for the node clustering task. For the baseline models, we trained the models in a

[1]https://github.com/NSLab-CUK/Unified-Graph-Transformer

	Chameleon		Squirrel		Film		Cornell		Texas		Wisconsin		Cora		Citeseer	
	C↓	Q↑	C↓	Q↑	C↓	Q↑	C↓	Q↑	C↓	Q↑	C↓	Q↑	C↓	Q↑	C↓	Q↑
GCN	0.46	0.25	0.27	0.73	0.59	0.11	0.75	0.00	0.61	0.16	0.59	0.18	0.16	0.68	0.17	0.63
GCNII	0.56	0.20	0.72	0.27	0.66	0.06	0.78	-0.05	0.80	-0.03	0.64	0.09	0.42	0.33	0.32	0.37
GIN	0.47	0.13	0.75	0.27	0.59	0.04	0.81	-0.10	0.90	-0.13	0.73	-0.02	0.20	0.63	0.24	0.56
GAT	0.66	0.09	0.77	0.25	0.71	0.02	0.85	-0.10	0.88	-0.18	0.82	-0.07	0.17	0.66	0.25	0.55
GATv2	0.72	0.06	0.77	0.20	0.70	0.06	0.80	-0.08	0.88	-0.13	0.72	-0.01	0.17	0.66	0.21	0.59
SAGE	0.77	0.03	0.75	0.25	0.72	0.00	0.90	-0.16	0.94	-0.15	0.83	-0.07	0.18	0.65	0.23	0.57
WRGAT	0.64	0.09	0.69	0.34	0.67	0.00	0.86	-0.15	0.92	-0.17	0.83	-0.12	0.23	0.60	0.28	0.53
WRGCN	0.64	0.10	0.68	0.34	0.67	0.00	0.88	-0.16	0.92	-0.17	0.79	-0.07	0.24	0.59	0.27	0.53
DGCN	0.75	0.04	0.78	0.26	0.75	0.03	0.80	-0.07	0.88	-0.15	0.75	-0.05	0.18	0.65	0.23	0.57
GT	0.78	0.02	0.78	0.22	0.80	-0.01	0.87	-0.15	0.92	-0.19	0.81	-0.11	0.19	0.64	0.26	0.54
SAN	0.71	0.01	0.58	0.42	0.42	0.01	0.89	-0.15	0.88	-0.12	0.79	-0.04	0.23	0.61	0.25	0.55
GPS	0.70	0.08	0.81	0.20	0.65	0.00	0.80	-0.09	0.73	-0.05	0.78	-0.05	0.38	0.46	0.50	0.29
SAT	0.80	-0.08	0.64	0.35	0.73	0.00	0.85	-0.07	0.7	0.02	0.81	-0.04	0.58	0.01	0.39	0.12
UGT(K)	0.12	**0.66**	0.24	**0.74**	0.44	0.32	0.52	0.10	0.80	-0.10	0.67	0.03	0.10	0.71	0.15	0.66
UGT	**0.11**	0.64	**0.21**	0.39	**0.28**	**0.50**	**0.28**	**0.47**	**0.33**	**0.46**	**0.27**	**0.52**	**0.09**	**0.76**	**0.04**	**0.78**

Table 1: The performance of node clustering task in terms of conductance (C) and modularity (Q) measurements on homophily and heterophily graphs. DGCN refers to the DeeperGCN model, and UGT(K) denotes that we passed the embeddings directly to the K-means algorithm after the pre-training phase.

	Brazil		Europe		USA	
	C↓	Q↑	C↓	Q↑	C↓	Q↑
GCN	0.71	-0.01	0.71	-0.07	0.47	0.16
GCNII	0.74	-0.02	0.76	-0.03	0.59	0.11
GIN	0.73	-0.03	**0.01**	0.01	**0.08**	0.10
GAT	0.68	-0.01	0.74	-0.05	0.69	0.03
GATv2	0.54	-0.01	0.61	-0.06	0.22	0.17
SAGE	0.55	-0.02	0.76	-0.06	0.68	0.04
WRGAT	0.74	-0.01	0.78	-0.05	0.71	0.03
WRGCN	0.72	0.00	0.79	-0.05	0.70	0.03
DGCN	0.46	0.00	0.55	-0.06	0.50	0.10
GT	0.61	-0.01	0.60	-0.05	0.31	0.12
SAN	0.60	-0.04	0.66	-0.06	0.42	0.15
GPS	0.70	0.03	0.70	-0.01	0.64	0.00
SAT	**0.43**	0.01	0.75	-0.06	0.72	0.00
UGT(K)	0.68	0.00	0.78	-0.05	0.13	0.22
UGT	0.51	**0.22**	0.51	**0.20**	0.34	**0.30**

Table 2: The performance of node clustering task on three Air-traffic networks.

supervised manner with a classification task. Then, we used learned representations as inputs for the K-means clustering algorithm. For our UGT model, we conducted clustering using both K-means and an end-to-end manner. Table 1 and Table 2 show the performance of node clustering in terms of conductance (C) and modularity (Q) measurements (Yang and Leskovec 2015). **(1)** UGT outperformed all baselines that ignore either local or global structures, e.g., GCN, WRGAT, and GT. Remarkably, the performance of our learned representations combined with K-mean also showed significant performance over the benchmark datasets. We suppose that as UGT could learn local and global structure and graph density, the representations could estimate the dense relations between nodes in communities. **(2)** Most of the recent graph transformers, such as SAN, SAT, Graphormer, and GPS, did not capture the density and connectivity information, only concerning higher-order neighbourhoods. This indicates that learning higher-order substructures could not help the model handle local relation density and graph partitioning.

Evaluation on node classification We show the results of the baseline and our models on the node classification task in Table 3 and Table 4. **(1)** Our UGT model with pre-training outperformed baseline models that overlook the long-term dependencies in most datasets, i.e., GCN, GAT, GATv2, and GT. We assume that the virtual edges allowed UGT to capture the long-term dependencies between structurally similar nodes, which have analogous roles and are significant for analyzing heterophily graphs. **(2)** UGT showed significant improvements in homophily graphs, i.e., Cora and Citeseer, in which nodes with the same labels tend to be adjacent. This indicates that UGT could not only capture local connectivity but also distinguish similar substructures (roles).

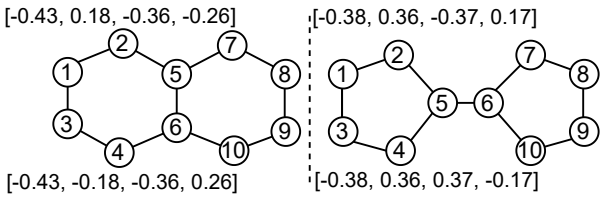
Evaluation on graph-level classification Table 5 shows the performance of the models on graph-level classification. For fair comparisons, we did not use the pre-training in the graph-level tasks, since graphs in the benchmarks have small scales. UGT exhibited competitive results compared to GNN variants and graph transformers, showing the effectiveness in capturing the global structure information in the whole graphs. This indicates that UGT could capture the connectivity in each individual substructure and then map unique substructures to different representations. For the MolHIV dataset, UGT slightly outperformed baselines, showing the ability of our proposed model to learn large-scale graphs.

	Chameleon	Squirrel	Film	Cornell	Texas	Wisconsin	Cora	Citeseer
GCN	66.25±1.77	50.92±1.02	28.15±1.37	55.55±9.29	48.89±6.47	58.40±6.49	87.33±1.71	72.05±1.22
GCNII	60.07±2.44	28.30±1.63	26.03±0.77	50.92±6.54	75.00±2.26	60.67±2.49	84.22±0.74	70.22±2.25
GIN	66.87±2.72	40.53±1.16	23.21±1.13	36.66±7.53	34.44±13.78	44.00±12.64	77.25±3.35	64.09±1.95
GAT	67.84±0.80	64.76±0.72	26.21±1.44	57.77±6.66	68.88±9.02	60.00±11.02	84.29±2.02	73.43±1.21
GATv2	62.20±2.11	50.80±3.01	25.73±2.11	56.66±8.88	63.33±9.68	55.20±4.66	85.77±1.27	72.95±1.87
SAGE	67.92±3.83	47.42±1.07	32.57±0.91	77.77±6.08	82.22±10.18	81.60±6.49	87.77±1.49	74.69±1.89
WRGAT	51.80±2.03	32.96±1.44	35.78±1.84	74.44±15.55	75.55±6.66	84.80±5.30	74.59±0.50	73.01±1.71
DGCN	57.53±2.82	34.96±1.04	31.68±1.50	63.33±10.30	76.67±8.16	72.80±8.54	86.59±0.85	73.61±0.62
GT	65.55±2.83	49.50±1.59	34.55±1.90	70.37±5.24	84.44±7.37	82.40±5.98	86.17±1.96	71.58±2.38
SAN	64.02±2.60	46.28±2.09	32.14±0.27	**79.62±2.61**	85.18±9.44	82.66±1.88	84.81±1.98	73.99±2.69
GPS	42.54±3.87	34.42±2.14	35.37±2.20	45.05±7.75	30.63±1.27	62.00±9.09	62.12±1.00	51.18±3.35
SAT	49.69±3.81	40.08±0.76	31.61±1.37	41.67±6.33	34.44±4.15	57.60±9.32	79.63±2.69	63.98±2.41
ANS-GT	54.60±1.02	35.80±1.17	**39.80±0.75**	59.80±2.71	63.80±5.04	**85.20±2.71**	88.00±0.63	75.00±0.89
GP	53.80±1.17	34.60±0.80	39.20±0.75	66.20±1.47	74.00±4.86	84.00±1.26	86.40±0.49	74.60±0.80
GD-WL	63.43±3.95	49.42±1.92	35.11±0.54	76.45±2.38	83.59±2.61	83.66±1.88	87.65±1.96	75.69±2.38
UGT(PT)	60.33±3.91	50.43 ±0.81	23.63 ±1.03	52.78±6.51	63.89 ±3.93	64.0±7.21	74.63±2.93	57.41±1.25
UGT	**69.78±3.21**	**66.96±2.49**	36.84±0.62	70.00±4.44	**86.67±8.31**	81.60±8.24	**88.74±0.60**	**76.08±2.50**

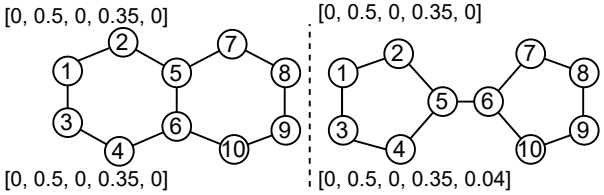
Table 3: The performance of node classification task (accuracy) on homophily and heterophily graphs. GP refers to the Graphormer model, and UGT(PT) denotes that we train UGT without pre-training.

	Brazil	Europe	USA
GCN	41.54±9.73	37.94±2.99	50.42±5.07
GCNII	24.35±3.62	22.78±5.37	28.29±4.19
GIN	50.76±7.84	42.56±9.67	50.75±9.77
GAT	58.46±7.84	54.35±3.40	46.38±11.41
GATv2	69.23±8.42	57.94±4.75	61.84±3.38
SAGE	66.15±7.84	60.16±4.59	60.00±6.60
WRGAT	55.38±3.07	51.28±3.62	56.97±3.54
DGCN	69.23±4.86	59.49±6.76	61.17±4.99
GT	63.07±11.30	62.56±9.53	64.37±2.67
SAN	61.53±10.87	**63.24±5.26**	35.29±3.43
GPS	52.31±8.97	46.00±5.15	42.52±3.85
SAT	66.15 ±6.15	57.43±6.19	65.04±4.06
ANS-GT	46.80±5.30	31.00±3.89	46.20±3.31
GP	42.00±3.20	29.80±1.17	45.80±3.66
GD-WL	68.97±9.59	60.68±2.41	63.46±0.79
UGT(PT)	76.92±4.71	44.94±4.86	64.71±3.19
UGT	**80.00±5.23**	56.92±6.36	**66.22±4.55**

Table 4: The performance of node classification task (accuracy) on three Air-traffic networks.



(a) Distinguishing non-isomorphic graphs with λ.



(b) Distinguishing non-isomorphic graphs with RWPE.

Figure 4: Distinguishing a Decalin graph (Left) and Bicyclopentyl graph (Right).

Evaluation on the power of model We conducted experiments to evaluate UGT on isomorphism testing for 1d-WL and 3d-WL benchmark datasets. Table 6 shows the performance on Graph8c and five SRGs datasets. **(1)** Our model could perfectly distinguish non-isomorphic graph pairs in Graph8c and SRGs. Most GNN variants could also distinguish graphs in Graph8c. However, as GNNs' capabilities are limited to the 1d-WL test, they failed to distinguish the graph pairs in SRGs. **(2)** The structural identity was effective to distinguish 1d-WL graph pairs in Graph8c, which is more powerful than ϵ of GIN. Since UGT determines substructures rooted in each node through structural identity (I), the learned representations thus could capture the surrounding substructures of each node. **(3)** To investigate the reasons why UGT is more powerful than baselines, we examined differences between Laplacian Eigenvectors and RWPE. Figure 4a presents how Laplacian Eigenvectors (λ) can distinguish a decalin and bi-cyclopentyl graph pair by assigning initial PEs to nodes '1' and '3'. At the first iteration, the input representation of UGT can be represented as $h_i^0 = \hat{x}_i^0 + W_1\lambda_i + b_1$ and assigns different representations from the first iteration. Thus, the output embeddings with structural identity ($h_i^{l+1} = FFN(I_i^l) + O_h^l(\cdot)$) ensure distinguishability to nodes. Thanks to the λ and I components,

	Enzymes (% ACC)	Proteins (% ACC)	NCI1 (% ACC)	NCI109 (% ACC)	MolHIV (% AUC)
GCN	18.24±2.05	59.23±0.62	68.24±2.38	67.09 ± 3.43	68.36±2.18
SAGE	21.46±4.32	62.79±1.38	64.36±2.82	64.47±2.41	67.13±3.54
GCNII	31.46±5.31	62.53±2.41	63.27±1.38	68.12±0.54	69.75±3.28
GIN	33.64±3.52	64.14 ± 2.05	66.72±5.32	68.44±1.89	69.42±3.47
GATv2	25.17±4.42	66.85±2.43	61.58±1.43	64.51±2.36	71.46±2.49
DeeperGCN	25.36±4.79	61.24±3.59	55.32±3.28	55.10±2.18	70.80±0.92
GT	41.67±6.67	77.25±3.83	69.77±1.40	69.66±0.06	74.86±1.24
SAN	22.50±0.83	68.47±0.90	59.31±3.47	57.30±8.20	73.75±1.65
SAT	50.85±3.66	62.91±0.76	54.99±0.26	56.04±0.06	**77 27±2.52**
GPS	62.71±8.64	53.75±6.20	**79.44±0.65**	**76.27±0.95**	74.01 ±3.18
UGT	**67.22±3.92**	**80.12±0.32**	77.55±0.16	75.45±1.26	76.24±0.29

Table 5: The performance on graph classification task in terms of ACC and AUC score.

	G8C	SRG1	SRG2	SRG3	SRG4
# Graphs	11K	15	10	4	28
# Pairs	61.8M	105	45	6	378
Types	1d-WL	3d-WL			
GCN	4,196	105	45	6	378
GA	1,827	105	45	6	378
GIN	559	105	45	6	378
ChebNet	44	105	45	6	378
PPGN	**0**	105	45	6	378
GNNML3	**0**	105	45	6	378
GT	**0**	**0**	**0**	**0**	2
GT w/o λ	6,157	105	45	6	378
SAN	5,819	105	45	6	378
SAT	286K	91	36	3	351
GPS	**0**	105	45	6	378
UGT w/o λ	355	105	45	6	378
UGT w/o I	36K	105	45	6	378
UGT	**0**	**0**	**0**	**0**	**0**

Table 6: The number of graph pairs that are undistinguished in Graph8c (G8C) and four SRGs graphs, including SRG1 (251256), SRG2 (261034), SRG3 (281264), and SRG4 (401224). An ideal graph representation model should not find any similar pairs.

UGT can distinguish almost non-isomorphic graph pairs up to 3d-WL testing. In contrast, RWPE needed a random walk with a length of five to distinguish the non-isomorphic graph pairs in Figure 4b. We suppose that in the SRGs, nodes are connected densely, which causes RWPE to fail to understand the substructures and connectivity in the graphs.

Sensitivity Analysis

We performed sensitivity analyses on the range of k-hop neighbourhoods and the ratio of virtual edges to real edges, as shown in Figure 5. **(1)** As k (i.e., the range of neighbourhood sampling) increased, the performance of node classification showed increasing tendencies across the three datasets. However, the trends were different for homophily and heterophily graphs. For homophily graphs (e.g., Cora), increasing k up to the 3-hop range affected model performance, but not beyond. In contrast, for heterophily and

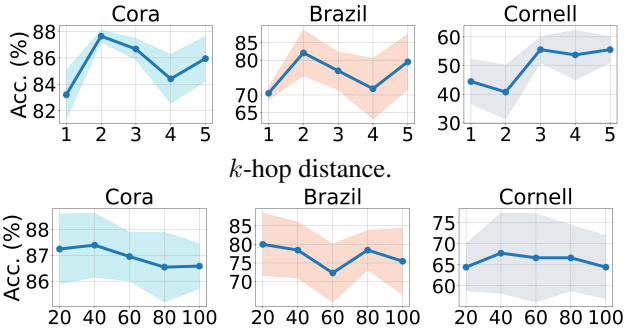


Figure 5: The performance of node classification task according to the sampling range of context nodes (Above) and the percentage of virtual edges over actual edges (Below).

sparse graphs, increasing the number of neighbourhoods provided more benefit. **(2)** Adding virtual edges could benefit more than increasing the number of k-hop neighbourhoods for the Cora dataset. We suppose that virtual edges could assist UGT in finding more distant nodes with structural similarity, and long-range dependencies could be effective for analyzing both homophily and heterophily graphs.

Conclusions

This study proposes a novel graph transformer model, UGT, to learn local and global structural features and integrate them into a unified representation. UGT captures long-range dependencies between nodes with similar roles using structural similarity-based sampling, discovers local connectivity using k-hop neighborhoods and structural identity, and unifies them by learning transition probabilities between nodes. Experimental results on various downstream tasks showed that UGT outperforms or is comparable to state-of-the-art baselines. Since our sampling technique uses the structural similarity in the whole graph, it can cause computational complexity when applied to large graphs. Thus, we plan to reduce the computational complexity by applying graph coarsening and within-community proximity.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1F1A1065516 and No. 2022K1A3A1A79089461) (O.-J.L.) and in part by the Research Fund, 2022 of The Catholic University of Korea (M-2023-B0002-00088) (O.-J.L.).

References

Balcilar, M.; Héroux, P.; Gaüzère, B.; Vasseur, P.; Adam, S.; and Honeine, P. 2021. Breaking the Limits of Message Passing Graph Neural Networks. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139 of *Proceedings of Machine Learning Research*, 599–608. Virtual Event: PMLR.

Brody, S.; Alon, U.; and Yahav, E. 2022. How Attentive are Graph Attention Networks? In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*. Virtual Event: OpenReview.net.

Chen, D.; O'Bray, L.; and Borgwardt, K. M. 2022. Structure-Aware Transformer for Graph Representation Learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, 3469–3489. Baltimore, Maryland, USA: PMLR.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, 1725–1735. PMLR.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS 2016)*, 3837–3845. Barcelona, Spain.

Dwivedi, V. P.; and Bresson, X. 2021. A Generalization of Transformer Networks to Graphs. In *Proceedings of the AAAI Workshop on Deep Learning on Graphs: Methods and Applications (AAAIw 2021)*.

Dwivedi, V. P.; Luu, A. T.; Laurent, T.; Bengio, Y.; and Bresson, X. 2022. Graph Neural Networks with Learnable Structural and Positional Representations. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*. Virtual Event: OpenReview.net.

Gutmann, M.; and Hyvärinen, A. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research*, 13: 307–361.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS 2017)*, 1024–1034. Long Beach, CA, USA.

Hoang, V. T.; Jeon, H.-J.; You, E.-S.; Yoon, Y.; Jung, S.; and Lee, O.-J. 2023a. Graph Representation Learning and Its Applications: A Survey. *Sensors*, 23(8).

Hoang, V. T.; Nguyen, S. T.; Lee, S.; Lee, J.; Nguyen, L. V.; and Lee, O.-J. 2023b. Companion Animal Disease Diagnostics Based on Literal-Aware Medical Knowledge Graph Representation Learning. *IEEE Access*, 11: 114238–114249.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019)*.

Jeon, H.; Choi, M.; and Lee, O.-J. 2022. Day-Ahead Hourly Solar Irradiance Forecasting Based on Multi-Attributed Spatio-Temporal Graph Convolutional Network. *Sensors*, 22(19): 7179.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. Toulon, France: OpenReview.net.

Kreuzer, D.; Beaini, D.; Hamilton, W. L.; Létourneau, V.; and Tossou, P. 2021. Rethinking Graph Transformers with Spectral Attention. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS 2021)*, 21618–21629. Virtual Event.

Lee, O.-J.; Jeon, H.; and Jung, J. J. 2021. Learning multi-resolution representations of research patterns in bibliographic networks. *Journal of Informetrics*, 15(1): 101126.

Lee, O.-J.; and Jung, J. J. 2020a. Story embedding: Learning distributed representations of stories based on character networks. *Artificial Intelligence*, 281: 103235.

Lee, O.-J.; and Jung, J. J. 2020b. Story Embedding: Learning Distributed Representations of Stories based on Character Networks (Extended Abstract). In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, 5070–5074. Yokohama, Japan: ijcai.org.

Li, G.; Xiong, C.; Thabet, A. K.; and Ghanem, B. 2020. DeeperGCN: All You Need to Train Deeper GCNs. *CoRR*, abs/2006.07739.

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *Proceedings of the ICML Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*.

Pei, H.; Wei, B.; Chang, K. C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. Addis Ababa, Ethiopia: OpenReview.net.

Rampásek, L.; Galkin, M.; Dwivedi, V. P.; Luu, A. T.; Wolf, G.; and Beaini, D. 2022. Recipe for a General, Powerful, Scalable Graph Transformer. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*.

Ribeiro, L. F. R.; Saverese, P. H. P.; and Figueiredo, D. R. 2017. *struc2vec*: Learning Node Representations from

Structural Identity. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, 385–394. Halifax, NS, Canada: ACM.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine*, 29(3): 93–106.

Suresh, S.; Budde, V.; Neville, J.; Li, P.; and Ma, J. 2021. Breaking the Limit of Graph Neural Networks by Improving the Assortativity of Graphs with Local Mixing Patterns. In *Proceedings of the 27th Conference on Knowledge Discovery and Data Mining (KDD 2021)*, 1541–1551. Virtual Event: ACM.

Tsitsulin, A.; Palowitch, J.; Perozzi, B.; and Müller, E. 2023. Graph Clustering with Graph Neural Networks. *Journal of Machine Learning Research*, 24: 127:1–127:21.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. Vancouver, BC, Canada: OpenReview.net.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. New Orleans, LA, USA: OpenReview.net.

Yang, J.; and Leskovec, J. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1): 181–213.

Ying, C.; Cai, T.; Luo, S.; Zheng, S.; Ke, G.; He, D.; Shen, Y.; and Liu, T. 2021. Do Transformers Really Perform Badly for Graph Representation? In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 28877–28888. Virtual Event.

Zhang, B.; Luo, S.; Wang, L.; and He, D. 2023. Rethinking the Expressive Power of GNNs via Graph Biconnectivity. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*. Kigali, Rwanda: OpenReview.net.

Zhang, Z.; Liu, Q.; Hu, Q.; and Lee, C. 2022. Hierarchical Graph Transformer with Adaptive Node Sampling. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2022)*.