# Improving the Expressiveness of $K$-hop Message-Passing GNNs by Injecting Contextualized Substructure Information

In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining

Tianjun Yao
tianjun.yao@mbzuai.ac.ae

MBZUAI

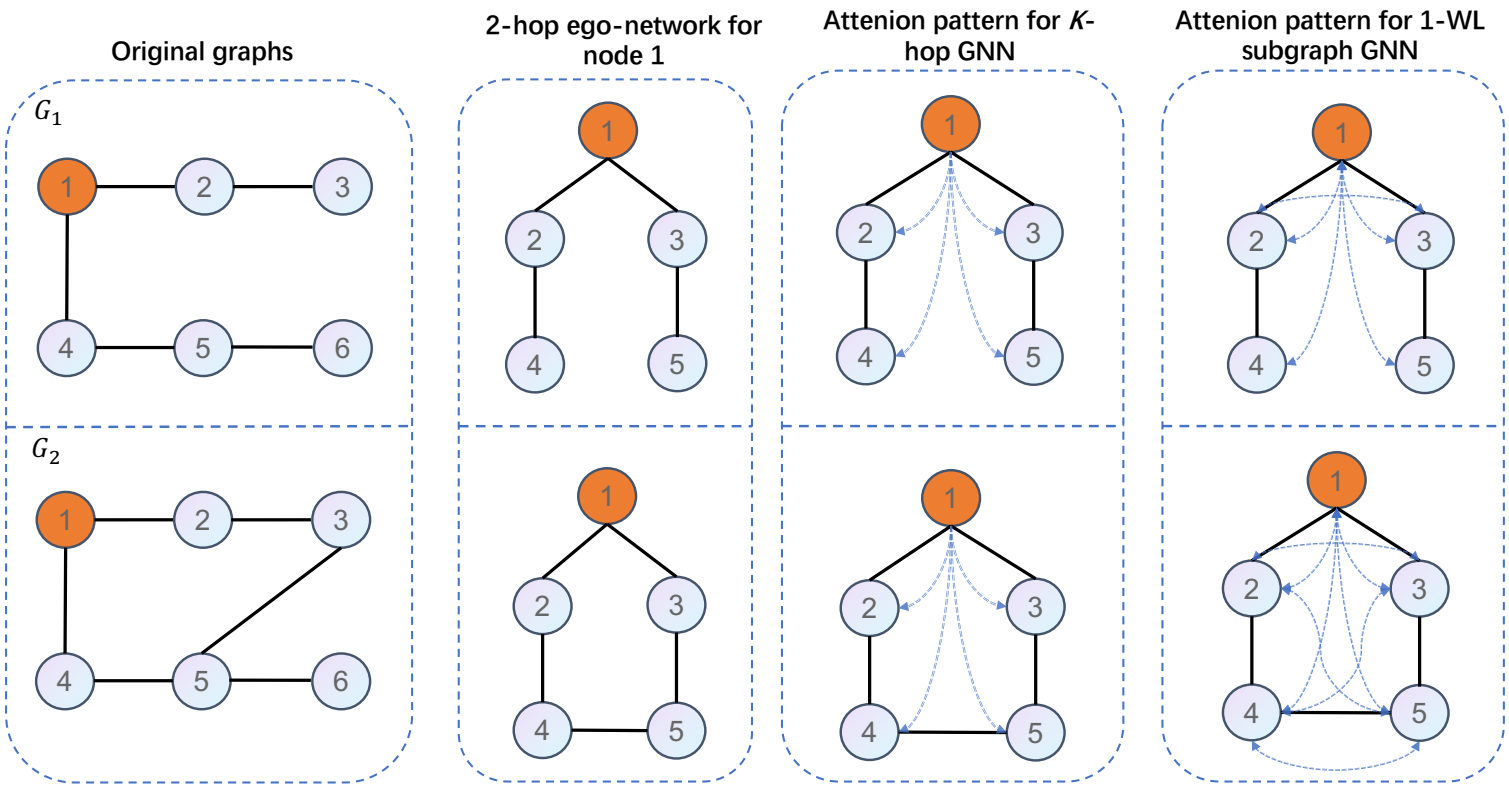
June 1, 2023

# Background

How to enhance the expressive power of graph neural networks is currently a active research area. Many methods have been proposed in the literatures.

- [1, 5, 12, 6, 7, 8] leverages positional and structural encodings to improve the expressive power.
- [10, 9, 4] uses higher-order message-passing schemes.
- [3, 9, 18, 17, 16] proposes to utilize subgraphs through GNNs.
- Some works proposes $K$-hop GNN models [2, 11, 13] that iteratively updates node representations by aggregating information from $K$-hop neighbors instead of direct neighbors.

# Motivation




Intuitively, 1-WL subgraph GNNs are more powerful than $K$-hop message-passing GNNs due to its pairwise attention mechanism.

Question: How to improve the expressive power of $K$-hop message-passing GNNs by mitigating this issue?

# Notations

| Symbol | Notation |
|---|---|
| $\{\}$ | sets |
| $\{\{\quad\}\}$ | multisets |
| $[n] := \{1, \cdots, n\}$ | index set |
| $G = (\mathcal{V}, \mathcal{E})$ | graph |
| $\mathcal{N}_G(u)$ | direct neighbors of node $u$ |
| $\mathcal{N}_G^k(u)$ | K-hop neighbors of node $u$ |
| $G_u^K$ | node-induced subgraph of root node $u$ |
| $\mathbf{p}$ | column vector |
| P | matrix |

# Proposed Method

## Definition: Substructure Encoding Function

A substructure encoding function $f : G_u^K \times G \to \mathcal{R}^d$ takes as input a $K$-hop subgraph rooted at node $u$ and graph $G$, and outputs an encoded $d$-dimensional features which can reflect the internal substructure of $G_u^K$.

- A proper substructure encoding function $f(\cdot)$ can capture the structural information of $G_u^K$, and mitigate the issues of hub-spoken attention patterns induced by $K$-hop message-passing GNNs.
- We resort to random walk to calculate the self-return probability for every node $u \in \mathcal{V}$. Intuitively, two nodes with different internal substructures would lead to different random walk patterns given sufficient steps of random walk.
- In addition to self-return probability, Central node to $k$-hop neighbors landing probability and Landing probability across $k$-hop neighbors are also designed to enrich the function.

# Substructure Encoding Function

$$
\begin{aligned}
f_1(G_u^K, G) &:= \left\{ H_{uu}^{(t)} \right\}_{t=1}^{l}, \\
f_2(G_u^K, G) &:= \mathrm{AGG}_k(\left\{ H_{ui}^{(t)} : dis(u,i) = k, k \in [K] \right\}_{t=1}^{l}), \\
f_3(G_u^K, G) &:= \mathrm{AGG}_k(\left\{ H_{i,j}^{(t)} : dis(i,u) = dis(j,u), k \in [K] \right\}_{t=1}^{l}), \\
f(G_u^K, G) &:= \mathrm{COMBINE}(f_i(G_u^K, G), i \in \{1,2,3\}).
\end{aligned}
\tag{1}
$$

In the above equations, $H^{(t+1)} = \widetilde{D}^{-1}\widetilde{A}H^{(t)}$ and $H^{(0)} = I$. $H^{(t)}$ is the $t^{th}$ step random-walk probability matrix.

## Theorem 1

Given two $n$-node $r$-regular graphs $G$ and $H$, let $3 \leq r < (2\log 2n)^{1/2}$ and $\epsilon$ be a fixed constant. For two $K$-hop ego-networks $G_u^K$ and $H_v^K$ with $K$ being at most $\left\lceil \left(\frac{1}{2} + \epsilon\right) \frac{\log 2n}{\log(r-1)} + 1 \right\rceil$, $2K$ steps of random walk is sufficient to discriminate the internal substructure of $G_u^K$ and $H_v^K$.

# Substructure-Enhanced 1-WL Algorithm

- *$K$-hop 1-WL Test*: A $K$-hop 1-WL color refinement algorithm iteratively refines node colors using all $k$-hops neighbors, where $k \in [K]$.
- *Subgraph 1-WL Test*: A subgraph 1-WL color refinement algorithm hashes the node-induced subgraph instead of direct neighbors.
- *SEK 1-WL Test*: SEK 1-WL updates node colors using both $K$-hop neighbors as well as the contextualized internal substructure information.

**Theorem 2**

SEK 1-WL test is strictly more powerful than $K$-hop 1-WL test and Subgraph 1-WL test.

# Practical Implementation

We provide an implementation of the SEK 1-WL algorithm, namely SEK-GNN. SEK-GNN achieves SOTA performance on a variety of datasets with significantly lower space and time complexity compared with 1-WL subgraph GNNs.

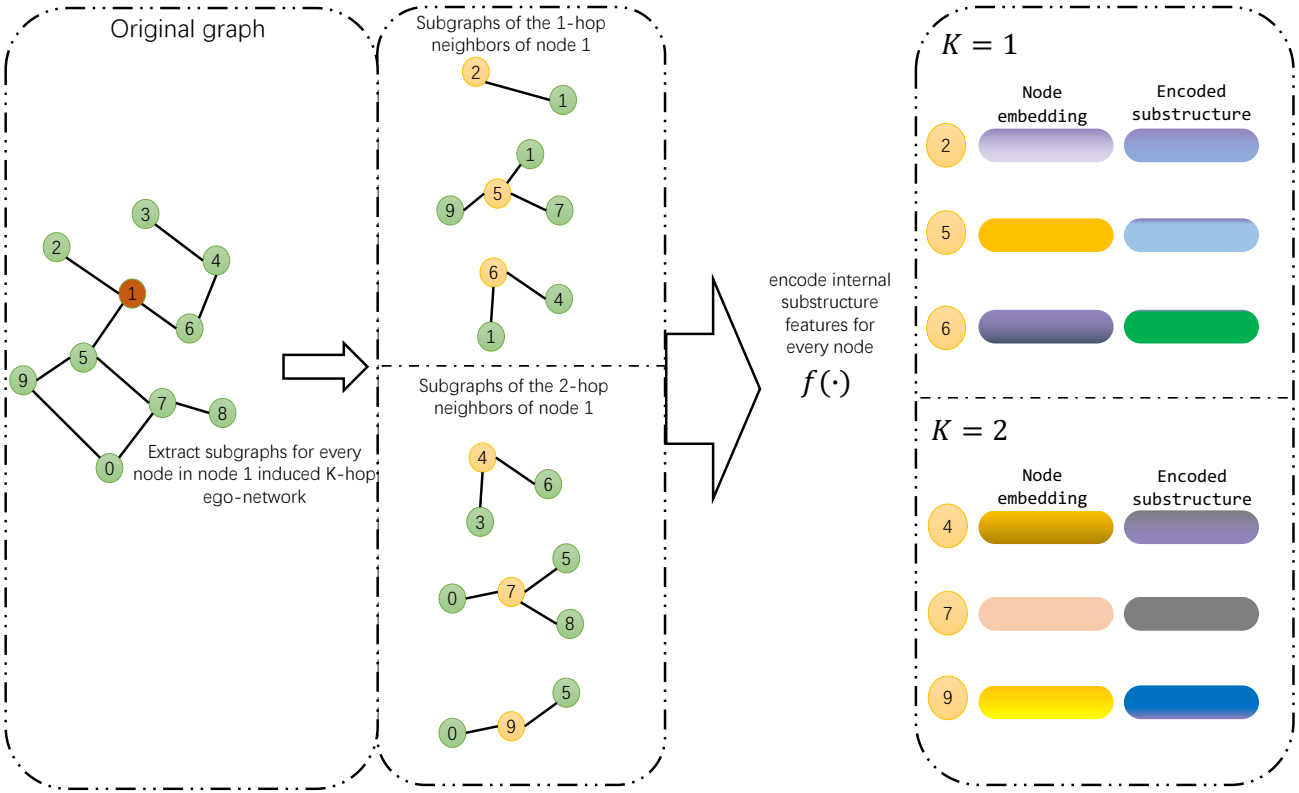
$$m_v^{l,k} = \mathrm{MESSAGE}_k^l \left( h_v^{l-1}, f\left(G_v^K, G\right), \left\{ \left\{ \left( h_u^{l-1}, f\left(G_u^K, G\right) \right) : u \in \mathcal{N}_G^k(v) \right\} \right\} \right),$$

$$h_v^{l,k} = \mathrm{UPDATE}_k^l \left( m_v^{l,k} \right),$$

$$h_v^l = \mathrm{COMBINE}^l \left( \left\{ \left\{ h_v^{l,k} : k \in [K] \right\} \right\} \right).$$

As shown in the equations, SEK-GNN can be easily implemented using a message-passing framework such as PyTorch Geometric.

# Model Architecture




SEK-GNN first extract subgraphs for central node($v_1$ in this case) and all contextualized nodes (up to $K$ hops), and utilize $f(\cdot)$ to encode the structural information, then use Eq. 8 to perform message-passing to update the node representation.

# Complexity Analysis

## Preprocessing

- *Space*. SEK-GNN only requires $\mathcal{O}(2m)$ space which is on par with standard GNNs. [a]
- *Time*. SEK-GNN requires $\mathcal{O}(lm)$ time to preprocess $\left\{H^{(t)}\right\}_{t=1}^{l}$ for a $l$-step random walk(The major bottleneck of our algorithm. However, it only performs one time, and $l$ is a small integer.)

[a] $m$ and $n$ are the number of edges and nodes in graph $\mathcal{G}$ respectively.

## Training/Inference

- *Space*. Requires $\mathcal{O}(n)$ space which is much more efficient than 1-WL subgraph GNN($\mathcal{O}(n^2)$).
- *Time*. Requires $\mathcal{O}(cn)$ time, also much faster than 1-WL subgraph GNN($\mathcal{O}(nm)$).

# Experiments on Synthetic Datasets

We perform experiments on both synthetic datasets and real-world datasets.

| Method | Graph Properties | | | Counting Substructures | | | |
|---|---|---|---|---|---|---|---|
| | Connect. | Diameter | Radius | Tri. | Tailed Tri. | Star | 4-Cycle |
| GIN | -1.9239 | -3.3079 | -4.7584 | 0.3569 | 0.2373 | 0.0224 | 0.2185 |
| PNA | -1.9395 | -3.4382 | -4.947 | 0.3532 | 0.2648 | 0.1278 | 0.243 |
| PPGN | -1.9804 | -3.6147 | -5.0878 | 0.0089 | 0.0096 | 0.0148 | 0.009 |
| GIN-AK+ | -2.7513 | -3.9687 | -5.1846 | 0.0123 | 0.0112 | 0.015 | 0.0126 |
| K-GIN+ | -2.1782 | **-3.9695** | -5.3088 | 0.1316 | 0.0813 | 0.0017 | 0.0916 |
| KP-GIN+ | **-4.4322** | -3.9361 | **-5.3345** | **0.0012** | **0.0016** | **0.0014** | 0.0067 |
| SEK-GIN | -2.6162 | -3.6101 | **-5.6821** | 0.0031 | 0.0058 | **0.0007** | **0.0032** |
| SEK-PPGN | **-4.9883** | **-4.4438** | -5.2273 | **0.0010** | **0.0021** | 0.0020 | **0.0063** |

Table: Simulation dataset results. Top two results are highlighted by **First** and **Second**.

# Experiments on TU Datasets

| Method | MUTAG | PTC-MR | PROTEINS | ENZYMES | BZR | COX2 | IMDB-B | IMDB-M |
|---|---|---|---|---|---|---|---|---|
| WL | 90.4 ± 5.7 | 59.9 ± 4.3 | 75.0 ± 3.1 | - | 78.5 ± 0.6 | 81.7 ± 0.7 | 73.8 ± 3.9 | 50.9± 3.8 |
| RetGK | 90.3 ± 1.1 | 62.5 ± 1.6 | 75.8 ± 0.6 | - | | - | 71.9 ± 1.0 | 47.7± 0.3 |
| GNTK | 90.0 ± 8.5 | **67.9 ± 6.9** | 75.6 ± 4.2 | - | 83.6 ± 2.9 | - | 76.9 ± 3.6 | - |
| WWL | 87.2 ± 1.5 | 66.3 ± 1.2 | 74.2 ± 0.5 | - | 84.4 ± 2.0 | 78.2 ± 0.4 | 74.3 ± 0.8 | - |
| FGW | 88.4 ± 5.6 | 65.3 ± 7.9 | 74.5 ± 2.7 | - | 85.1 ± 4.1 | 77.2 ± 4.8 | 63.8 ± 3.4 | - |
| DGCNN | 85.8 ± 1.7 | 58.6 ± 2.5 | 75.5 ± 0.9 | 38.9±5.7 | - | - | 70.0 ± 0.9 | 47.8 |
| CapsGNN | 86.6 ± 6.8 | 66.0 ± 1.8 | 76.2 ± 3.6 | - | - | - | 73.1 ± 4.8 | - |
| GraphSAGE | 85.1 ± 7.6 | 63.9 ± 7.7 | 75.9 ± 3.2 | - | - | - | 72.3 ± 5.3 | - |
| GIN | 89.4 ± 5.6 | 64.6 ± 7.0 | 75.9 ± 2.8 | 59.6±4.5 | - | - | 75.1 ± 5.1 | **52.3± 2.8** |
| GIN-AK+ | 91.3± 7.0 | **68.20± 5.6** | **77.1 ± 5.7** | - | - | - | 75.6 ± 3.7 | - |
| GraphSNN | 91.57 ± 2.8 | 66.70 ± 3.7 | 76.83 ± 2.5 | **61.7±3.4** | **88.69 ± 3.2** | **82.86 ± 3.1** | **77.86 ± 3.6** | - |
| KP-GIN | **92.2± 6.5** | 66.80 ± 6.8 | 75.80 ± 4.6 | - | - | - | 76.6 ± 4.2 | - |
| SEK-GIN | **92.2± 6.2** | 66.86± 7.2 | **78.32 ± 2.7** | **61.85±5.1** | **89.65 ± 3.8** | **85.89±4.4** | **76.9 ± 3.8** | **53.80 ± 3.2** |
| †GIN-AK+ | 95.1± 6.1 | 74.1± 5.9 | 78.9 ± 5.4 | - | - | - | 77.3 ± 3.1 | - |
| †GraphSNN | 94.70 ± 1.9 | 70.58 ± 3.1 | 78.42 ± 2.7 | - | **91.12 ± 3.0** | **86.28 ± 3.3** | 78.51 ± 2.8 | - |
| †KP-GIN | **96.1± 4.6** | **76.20 ± 4.5** | **79.50 ± 4.4** | - | - | - | **80.7 ± 2.6** | - |
| †SEK-GIN | **96.9± 3.6** | **75.71± 6.4** | **81.70 ± 2.5** | **66.12±5.4** | **93.36± 2.3** | **89.73±2.5** | **80.3± 3.2** | **56.13 ± 3.0** |

Table: Evaluation result on TU dataset using two different evaluation settings. The first one follows [15], and the second follows [14]. We use † to denote the second setting. Top two results are highlighted by **First** and **Second**.

# Experiments on QM9 Dataset

| Target | DTNN | MPNN | Deep LRP | PPGN | Nested 1-2-3-GNN | KP-GIN+ | SEK-GIN |
|---|---|---|---|---|---|---|---|
| $\mu$ | **0.244** | 0.358 | 0.364 | **0.231** | 0.433 | 0.367 | 0.358 |
| $\alpha$ | 0.95 | 0.89 | 0.298 | 0.382 | 0.265 | **0.242** | **0.228** |
| $\varepsilon_{\text{HOMO}}$ | 0.00388 | 0.00541 | 0.00254 | 0.00276 | 0.00279 | **0.00247** | **0.00106** |
| $\varepsilon_{\text{LUMO}}$ | 0.00512 | 0.00623 | 0.00277 | 0.00287 | 0.00276 | **0.00238** | **0.00229** |
| $\Delta\varepsilon$ | 0.0112 | 0.0066 | 0.00353 | 0.00406 | 0.0039 | **0.00345** | **0.00335** |
| $\langle R^2 \rangle$ | 17 | 28.5 | 19.3 | **16.7** | 20.1 | **16.49** | 16.91 |
| ZPVE | 0.00172 | 0.00216 | 0.00055 | 0.00064 | **0.00015** | 0.00018 | **0.00013** |
| $U_0$ | 2.43 | 2.05 | 0.413 | 0.234 | 0.205 | **0.0728** | **0.0587** |
| $U$ | 2.43 | 2 | 0.413 | 0.234 | 0.2 | **0.0553** | **0.0672** |
| $H$ | 2.43 | 2.02 | 0.413 | 0.229 | 0.249 | **0.0575** | **0.073** |
| $G$ | 2.43 | 2.02 | 0.413 | 0.238 | 0.253 | **0.0526** | **0.0592** |
| $C_v$ | 0.27 | 0.42 | 0.129 | 0.184 | **0.0811** | 0.0973 | **0.0924** |

Table: Evaluation result on QM9, Top two results are highlighted by **First** and **Second**.

# Ablation Study on SEK-GIN

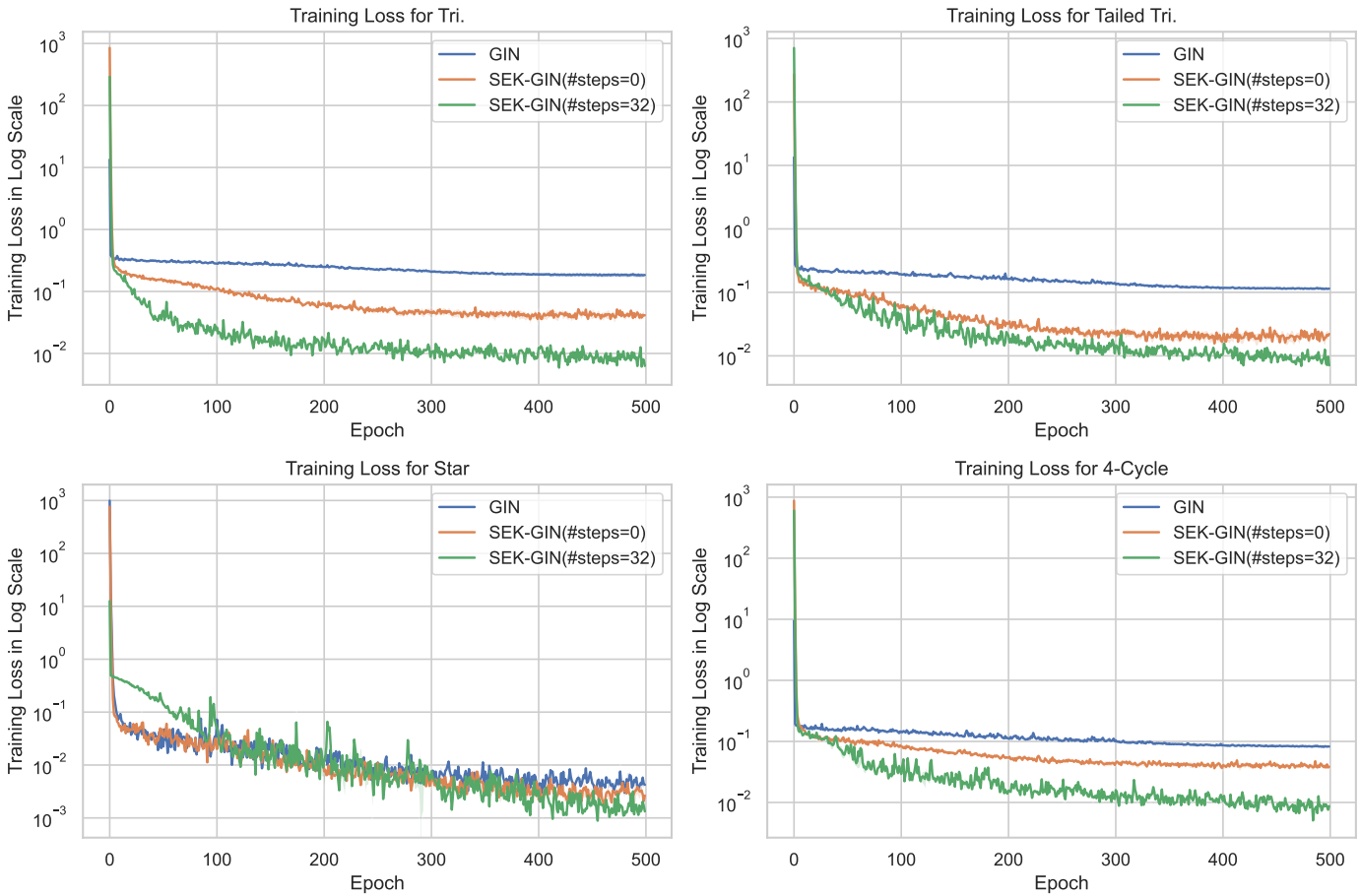
| Method | Substructure counting | | | Graph property regression | | | |
|---|---|---|---|---|---|---|---|
| | Tri. | Tailed Tri. | Star | 4-cycle | Connect. | Diameter | Radius |
| SEK-GIN(#steps=0) | 0.1529 | 0.0891 | 0.0011 | 0.0948 | -2.215 | -2.647 | -4.823 |
| SEK-GIN(#steps=8) | 0.0043 | **0.0059** | 0.0008 | 0.0039 | -2.439 | -3.078 | -4.915 |
| SEK-GIN(#steps=16) | 0.0034 | 0.0061 | **0.0007** | **0.0037** | -2.555 | -3.268 | -4.902 |
| SEK-GIN(#steps=24) | 0.0034 | 0.0068 | 0.0007 | 0.0043 | **-2.765** | -3.423 | -5.005 |
| SEK-GIN(#steps=32) | **0.0033** | 0.0066 | 0.0008 | 0.0043 | -2.663 | **-3.591** | **-5.103** |

Table: Ablation study on synthetic datasets using SEK-GIN

| Method | IMDB-BINARY | PROTEINS | BZR | MUTAG |
|---|---|---|---|---|
| SEK-GIN(#steps=0) | 0.781±0.037 | 0.790±0.030 | 0.906±0.021 | 0.936±0.052 |
| SEK-GIN(#steps=8) | 0.791±0.036 | 0.796±0.035 | 0.911±0.030 | 0.952±0.039 |
| SEK-GIN(#steps=16) | **0.792±0.043** | 0.803±0.034 | 0.911±0.019 | 0.952±0.039 |
| SEK-GIN(#steps=24) | 0.785±0.041 | 0.803±0.029 | 0.912±0.019 | 0.952±0.029 |
| SEK-GIN(#steps=32) | 0.787±0.034 | **0.808±0.027** | **0.921±0.022** | **0.958±0.033** |

Table: Ablation study on TU datasets using SEK-GIN

# Training Curves on SubstructureCount Dataset




As illustrated in the figure, incorporating substructure information brings performance gains for all four tasks.

# Summarization

Our main contribution is summarized as follows:

- We propose a general notion of *substructure encoding function* by establishing connections between $K$-hop message-passing GNNs and 1-WL subgraph GNNs.
- We propose an instantiation of the substructure encoding function by leveraging multi-step random walk. Our proposed structural encoding scheme is effective, parallelizable with theoretical guarantees to extract structural information of subgraphs.
- We propose SEK 1-WL test, which is provably more powerful than Subgraph 1-WL test and $K$-hop 1-WL test, and is not less powerful than 3-WL test.
- We propose an implementation of SEK 1-WL test, namely SEK-GNN, which is easy to implement, and can be naturally incorporated into message passing framework with parallelizability and effectiveness. Furthermore, SEK-GNN is able to achieve SOTA performance in a variety of datasets with significantly lower space complexity than 1-WL subgraph GNNs.

# Bibliography I


[1] Ralph Abboud et al. "The surprising power of graph neural networks with random node initialization". In: *arXiv preprint arXiv:2010.01179* (2020).

[2] Sami Abu-El-Haija et al. "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing". In: *international conference on machine learning*. PMLR. 2019, pp. 21–29.

[3] Beatrice Bevilacqua et al. "Equivariant subgraph aggregation networks". In: *arXiv preprint arXiv:2110.02910* (2021).

[4] Cristian Bodnar et al. "Weisfeiler and Lehman go cellular: CW networks". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 2625–2640.

[5] Giorgos Bouritsas et al. "Improving graph neural network expressivity via subgraph isomorphism counting". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

# Bibliography II


[6] Vijay Prakash Dwivedi et al. "Graph neural networks with learnable structural and positional representations". In: *arXiv preprint arXiv:2110.07875* (2021).

[7] Devin Kreuzer et al. "Rethinking graph transformers with spectral attention". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 21618–21629.

[8] Derek Lim et al. "Sign and Basis Invariant Networks for Spectral Graph Representation Learning". In: *arXiv preprint arXiv:2202.13013* (2022).

[9] Haggai Maron et al. "On learning sets of symmetric elements". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6734–6744.

# Bibliography III


[10] Christopher Morris et al. "Weisfeiler and leman go neural: Higher-order graph neural networks". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 4602–4609.

[11] Giannis Nikolentzos, George Dasoulas, and Michalis Vazirgiannis. "k-hop graph neural networks". In: *Neural Networks* 130 (2020), pp. 195–205.

[12] Omri Puny, Heli Ben-Hamu, and Yaron Lipman. "Global attention improves graph networks generalization". In: *arXiv preprint arXiv:2006.07846* (2020).

[13] Guangtao Wang et al. "Multi-hop attention graph neural network". In: *arXiv preprint arXiv:2009.14332* (2020).

[14] Asiri Wijesinghe and Qing Wang. "A New Perspective on" How Graph Neural Networks Go Beyond Weisfeiler-Lehman?"" In: *International Conference on Learning Representations*. 2021.

# Bibliography IV


[15] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[16] Jiaxuan You et al. "Identity-aware graph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 10737–10745.

[17] Muhan Zhang and Pan Li. "Nested graph neural networks". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 15734–15747.

[18] Lingxiao Zhao et al. "From stars to subgraphs: Uplifting any GNN with local structure awareness". In: *arXiv preprint arXiv:2110.03753* (2021).

Thank You for Your Attention!