Structure-Aware Transformer for Graph Representation Learning

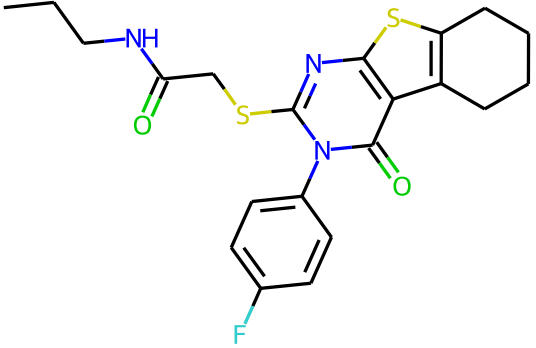
Dexiong Chen[*] [1,2] Leslie O'Bray[*] [1,2] Karsten Borgwardt[1,2]

*: Equal contribution

D BSSE

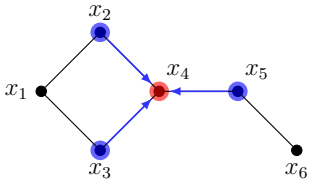
ETH zürich

Graph representation learning



How to define a (parameterized) function $\varphi_\theta : \mathcal{G} \to \mathbb{R}^d$ for graph representations that fully explores the information about the **graph structure**?

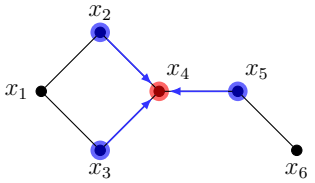
Message passing graph neural networks use neighborhood aggregation



Limitations

- Modeling **long-range dependencies**
- Strong **structural inductive bias**
- **Over-smoothing**
- **Over-squashing**

Message passing graph neural networks use neighborhood aggregation



Limitations

- Modeling **long-range dependencies**
- Strong **structural inductive bias**
- **Over-smoothing**
- **Over-squashing**

$\Rightarrow$ We need architectures beyond neighborhood aggregation!

Graph transformers could address some limitations of MPGNNs

Key idea of existing work:

- Encode the **structural or positional relationships** between nodes into the Transformer architecture.

Graph transformers could address some limitations of MPGNNs

Key idea of existing work:

- Encode the **structural or positional relationships** between nodes into the Transformer architecture.

Our contribution

- Generalize self-attention to account for local structures by extracting a **subgraph representation** rooted at each node.
- Resulting framework can leverage **any GNN** to extract subgraph representations and create **structure-aware node representations**.
- Empirically outperforms the base GNN $\Rightarrow$ an effortless enhancer of any GNN.

From attention to structure-aware attention

self-attention | **structure-aware self-attention**

$$\text{Attn}(x_v) = \sum_{u \in V} \frac{\kappa_{\exp}(x_v, x_u)}{\sum_{w \in V} \kappa_{\exp}(x_v, x_w)} f(x_u) \quad \Rightarrow \quad \text{SA-attn}(v) = \sum_{u \in V} \frac{\kappa_{\text{graph}}(S_G(v), S_G(u))}{\sum_{w \in V} \kappa_{\text{graph}}(S_G(v), S_G(w))} f(x_u)$$

- Consider a graph $G = (V, E, X)$ with node attributes $X = (x_v)_{v \in V}$.
- Self-attention as **kernel smoothing** with

$$\kappa_{\exp}(x, x') = e^{\frac{\langle W_Q x, W_K x' \rangle}{\sqrt{d_{out}}}} \qquad f(x) = W_V x \in \mathbb{R}^{d_{out}}$$

- $S_G(v)$ is a subgraph rooted at node v. κ_{graph} measures similarity between graphs.
- A wide class of **expressive** and **tractable** kernels:

$$\kappa_{\text{graph}}(S_G(v), S_G(u)) = \kappa_{\exp}(\underbrace{\varphi(v, G)}_{\text{structure extractor}}, \varphi(u, G))$$

[Tsai et al., 2019]

Structure extractors

k-subtree GNN extractor

- Use a GNN to extract the representation of the k-subtree structure:

$$\varphi(u, G) := \text{GNN}_G^{(k)}(u). \tag{1}$$

- Any existing GNN can be used.

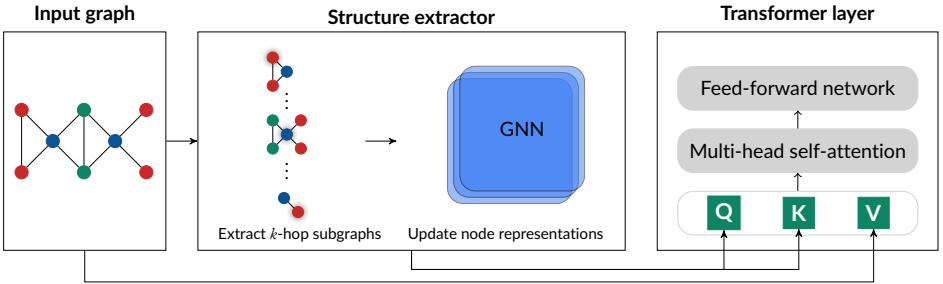
k-subgraph GNN extractor

- Extract the k-hop subgraph $\mathcal{N}_k(u)$ rooted at node u.
- Use a GNN to compute the representation of the k-hop subgraph:

$$\varphi(u, G) := \sum_{v \in \mathcal{N}_k(u)} \text{GNN}_{\mathcal{N}_k(u)}^{(k)}(v). \tag{2}$$

- More **expressive** but computationally more **expensive**.

Structure-aware transformer



$\Rightarrow$ SAT consists of multiple Transformer layers.

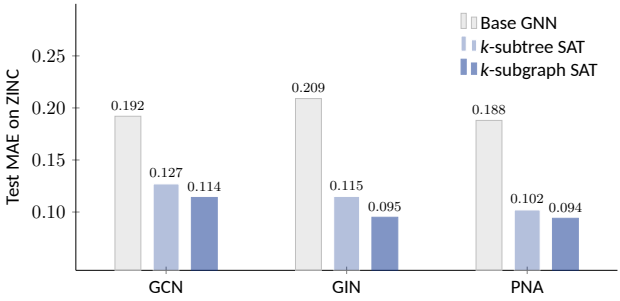
Experiments: SAT achieves SOTA results on several datasets

	ZINC ↓	CLUSTER ↑	PATTERN ↑
# graphs	12,000	12,000	14,000
Avg. # nodes	23.2	117.2	118.9
Avg. # edges	49.8	4,303.9	6,098.9
Metric	MAE	Accuracy	Accuracy
GIN	0.387±0.015	64.716±1.553	85.590±0.011
GAT	0.384±0.007	70.587±0.447	78.271±0.186
PNA	0.188±0.004	67.077±0.977*	86.567±0.075
Transformer+RWPE	0.310±0.005	29.622±0.176	86.183±0.019
SAN	0.139±0.006	76.691±0.650	86.581±0.037
Graphormer	0.122±0.006	–	–
k-subtree SAT	0.102±0.005	77.751±0.121	**86.865±0.043**
k-subgraph SAT	**0.094±0.008**	**77.856±0.104**	86.848±0.037

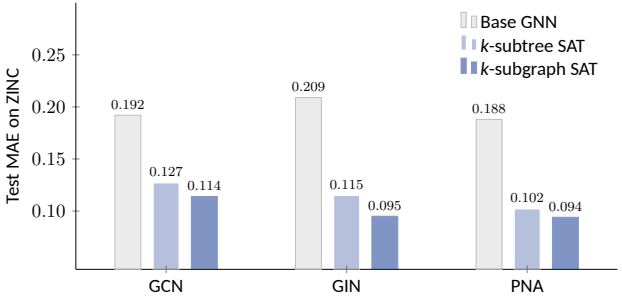
	OGBG-PPA ↑	OGBG-CODE2 ↑
# graphs	158,100	452,741
Avg. # nodes	243.4	125.2
Avg. # edges	2,266.1	124.2
Metric	Accuracy	F1 score
GCN-Virtual Node	0.6857±0.0061	0.1595±0.0018
GIN-Virtual Node	0.7037±0.0107	0.1581±0.0026
Transformer	0.6454±0.0033	0.1670±0.0015
GraphTrans	–	0.1830±0.0024
k-subtree SAT	**0.7522±0.0056**	**0.1937±0.0028**

[Xu et al., 2019, Veličković et al., 2018, Corso et al., 2020, Kreuzer et al., 2021, Ying et al., 2021]

SAT empirically always improved upon base GNNs



SAT empirically always improved upon base GNNs



- More results on hyperparameter studies and model interpretation?
- Please consult our github repo and come to our poster!

`https://github.com/BorgwardtLab/SAT`

References I

G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković. Principal neighbourhood aggregation for graph nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

D. Kreuzer, D. Beaini, W. L. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. Salakhutdinov. Transformer dissection: A unified understanding of transformer's attention via the lens of kernel. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.