GraphGPS : General Powerful Scalable Graph Transformers

One-time paper review

ii tae jeong
jeongiitae6@gmail.com
seoul , south korea

Outline

Preliminary

- ➢ Message passing GNN vs. Graph Transformers
- ➢ isomorphism test
- ➢ Linear Transformers

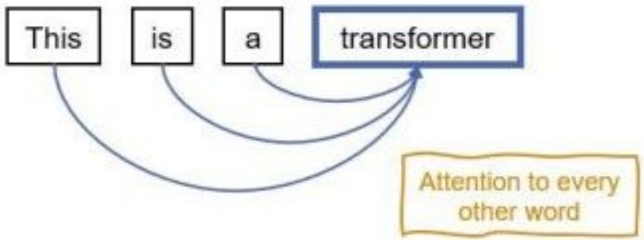
Novelty

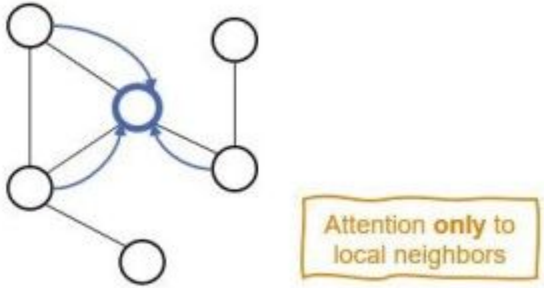
Recipes

Open Discussion

Preliminary

Graph Transformer



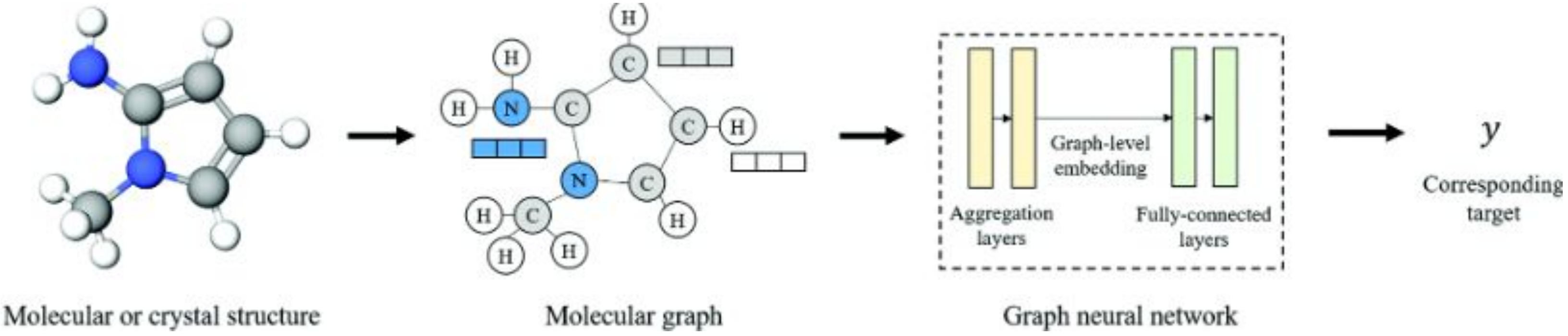


<Sequence>
This is a transformer

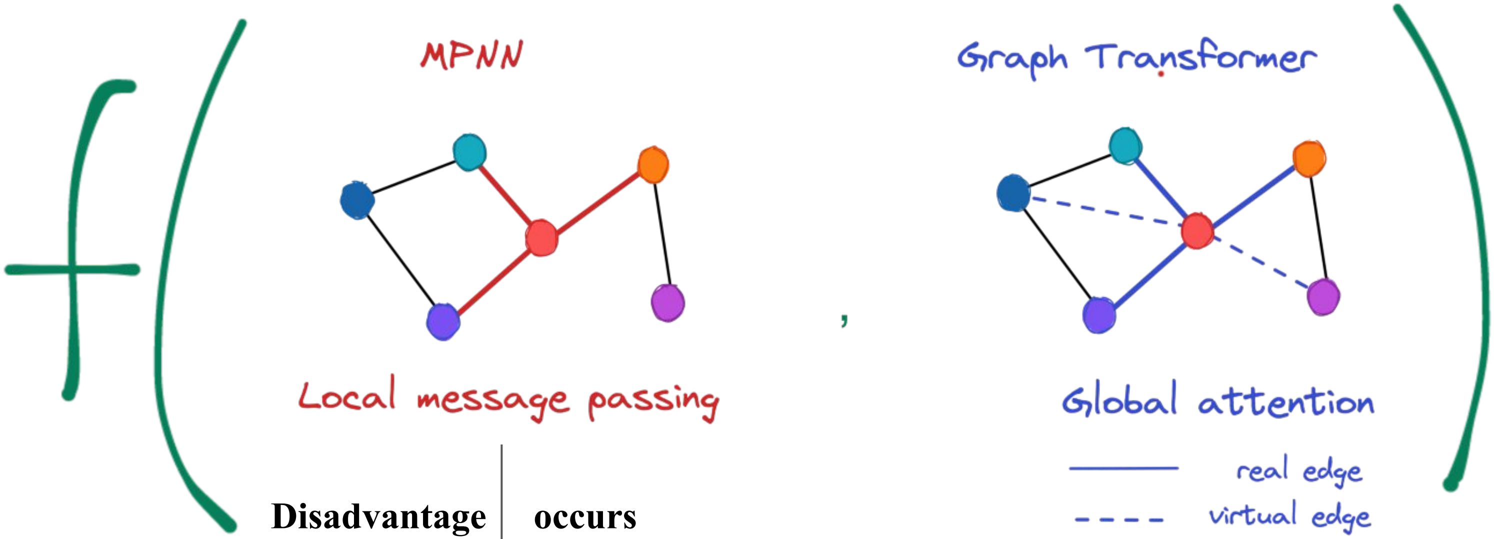
v/s

<Multi-set>
{This, is, a, transformer}

why important these PE&SE information?



MPNN vs. Graph Transformer

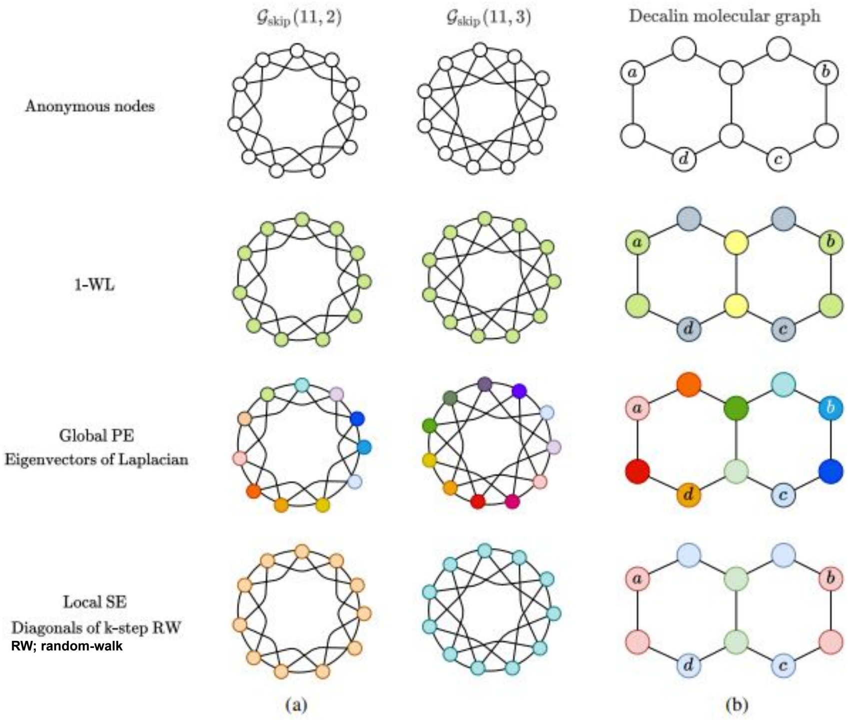


Disadvantage occurs

- **over-smoothing** (increasing the number of GNN layers, the features tend to converge to the same value),
- **over-squashing** (losing information when trying to aggregate messages from many neighbors into a single vector)
- poor capturing of long-range dependencies which is noticeable already on small but sparse molecular graphs.

Keypoint

Transformer architecture had been mainly used to text data at NLP industry. It was able to distinguish by specifying the data. **But GNN couldn't it since the node ordering.** so we need to that tools for specifying what the data identifiability.



- node) CSL(Circular Skip Link) graph for isomorphic task which is capture differentiate between two potential links

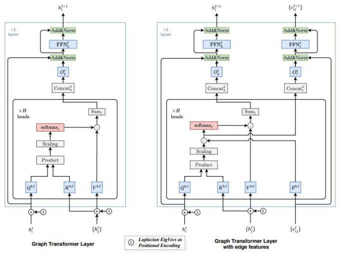
task → the feature embeddings of the two graphs **which are the hash function outputs of the collection of node colors are different**, thus making the task to distinguish the graphs successful

- edge) Decalin molecular graph, the node a is isomorphic to node b, and so is the node c to node d.

task → Identifying a potential link between the node-set(a,d) and (b,d), **the combination of the node colors of the node-sets will produce the same embedding for the two links.**

Graph Transformer roadmap

Laplacian eigenvector



shortest path distance

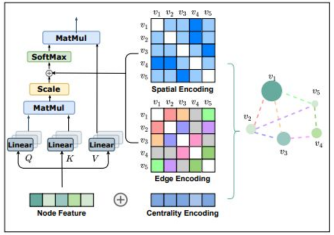
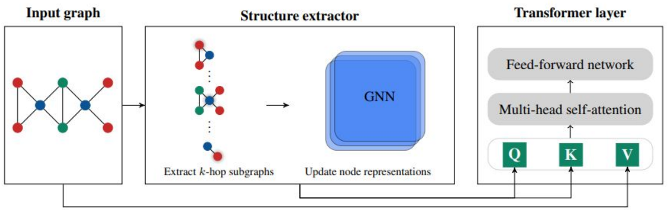


Figure 1: An illustration of our proposed centrality encoding, spatial encoding, and edge encoding in Graphormer.

aggregating a k-hop subgraph around each node



2020, dec

2021, Jun

2021, Nov

2022, Jan

2022, Jun

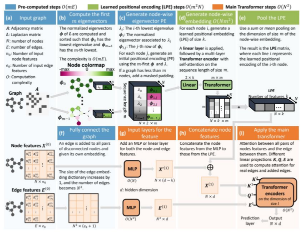


Figure 1: The proposed SAN model with the node LPE, a generalization of Transformers to graphs.

Laplacian eigenvalues to re-weight attention

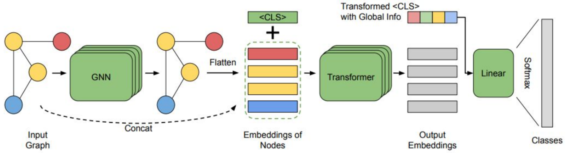
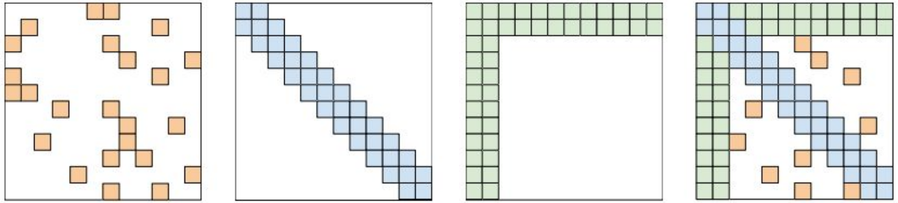


Figure 1: Architecture of GraphTrans. A standard GNN submodule learns local, short-range structure, then a global Transformer submodule learns global, long-range relationships.

Run a GT after passing a graph through a GNN

Bigbird



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Figure 1: Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with $r = 2$, (b) sliding window attention with $w = 3$ (c) global attention with $g = 2$. (d) the combined BIGBIRD model.

- A set of g global tokens attending on all parts of the sequence.
- All tokens attending to a set of w local neighboring tokens.
- All tokens attending to a set of r random tokens.

$$\text{ATTN}_D(\boldsymbol{X})_i = \boldsymbol{x}_i + \sum_{h=1}^{H} \sigma\left(Q_h(\boldsymbol{x}_i) K_h(\boldsymbol{X}_{N(i)})^T\right) \cdot V_h(\boldsymbol{X}_{N(i)})$$

Let N(i) denote the out-neighbors set of node i in D, then the i^th output vector of the generalized attention mechanism

Model	MLM	SQuAD	MNLI
BERT-base	64.2	88.5	83.4
Random (R)	60.1	83.0	80.2
Window (W)	58.3	76.4	73.1
R + W	62.7	85.1	80.5

Table 1: Building block comparison @512

Performer

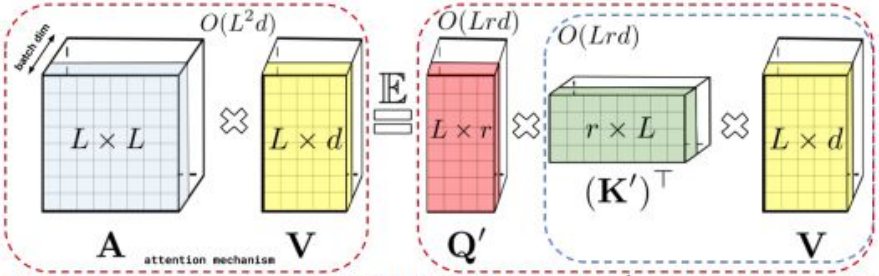


Figure 1: Approximation of the regular attention mechanism **AV** (before $\mathbf{D}^{-1}$-renormalization) via (random) feature maps. Dashed-blocks indicate order of computation with corresponding time complexities attached.

softmax kernel approximation

Novelty

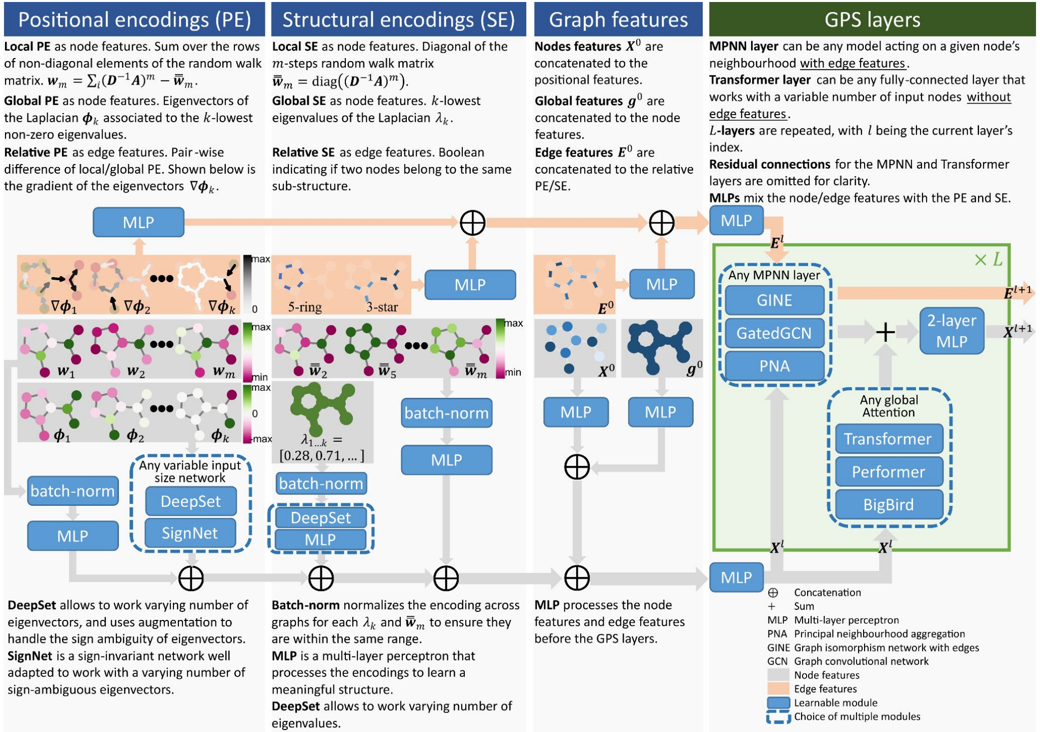
Novelty

1. Scalable (linear global attention)
   → efficient implementation at graph transformer speed) 400% faster without explicit edge features within the attention module.

2. Generalization (isomorphism)
   → using the positional , structural encoding tricks and ablation study

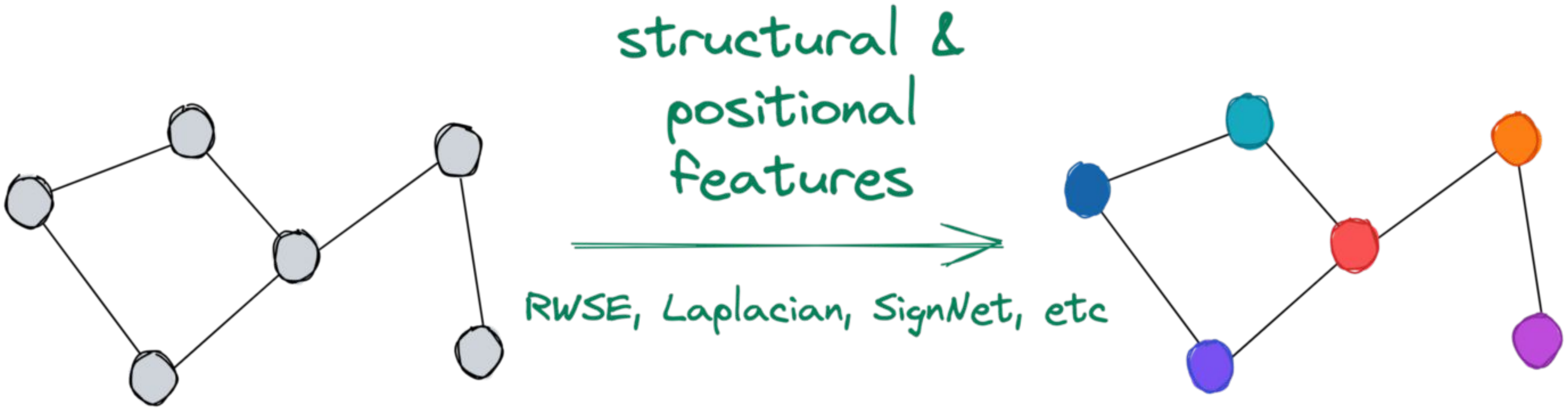
3. Powerful
   → SOTA

Recipes



1. Node identification through positional and structural encodings. After analyzing many recently published methods for adding positionality in graphs, we found they can be broadly grouped into 3 buckets: local, global, and relative. Such features are provably powerful and help to overcome the notorious 1-WL limitation.
2. Aggregation of node identities with original graph features — those are your input node, edge, and graph features.
3. Processing layers (GPS layers) — how we actually process the graphs with constructed features, here combine both local message passing (any MPNNs) and global attention models (any graph transformer)

component 1. SE & PE encoding



Positional encodings   where am I?

Local PE (node)

* Distance to cluster center
* Sum of non-diagonal elements in m-step RW

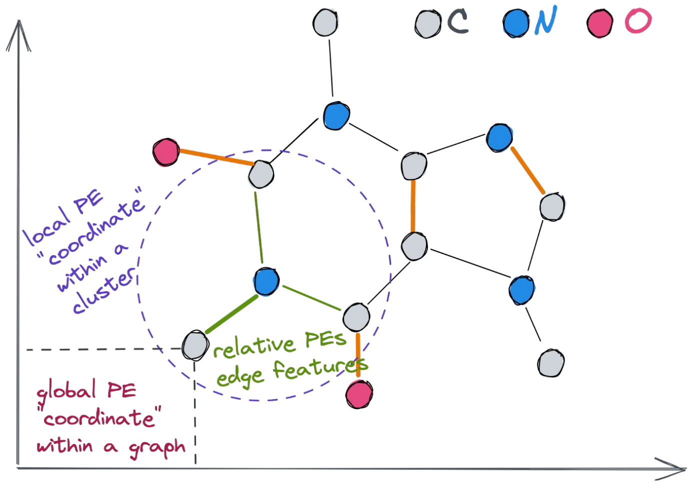
Global PE (node)

* Eigenvectors of A, L, or distance matrix
* Distance to graph's centroid
* Unique ID for each CC

Relative PE (edge)

Pair-wise dists from
* Heat kernels
* Random Walks
* Green's function
* Graph geodesic

* Gradient of eigenvectors



Structural encodings — what does my neighborhood look like?

Local SE (node)	Global SE (graph)	Relative SE (edge)
* Node degree	* Eigenvalues of A, L	* Gradient of any Local SE
* RW diagonals	* Graph diameter, girth, degree, #CC	* Gradient of sub-structure enumeration
* Ricci curvature		
* Enumerate substructures (triangles, rings)		

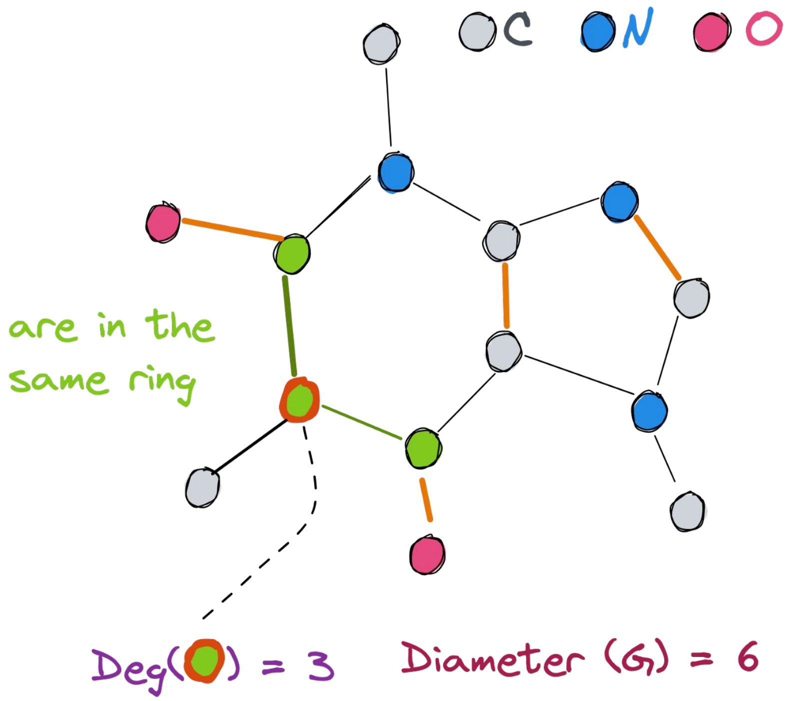


Table 1: The proposed categorization of positional encodings (PE) and structural encodings (SE). Some encodings are assigned to multiple categories in order to show their multiple expected roles.

Encoding type	Description	Examples
Local PE *node features*	Allow a node to know its position and role within a local cluster of nodes. *Within a **cluster**, the closer two nodes are to each other, the closer their local PE will be, such as the position of a word in a sentence (not in the text).*	• Sum each column of non-diagonal elements of the m-steps random walk matrix. • Distance between a node and the centroid of a cluster containing the node.
Global PE *node features*	Allow a node to know its global position within the graph. *Within a **graph**, the closer two nodes are, the closer their global PE will be, such as the position of a word in a text.*	• Eigenvectors of the Adjacency, Laplacian [14, 34] or distance matrices. • Distance from the graph's centroid. • Unique identifier for each connected component of the graph.
Relative PE *edge features*	Allow two nodes to understand their distances or directional relationships. *Edge embedding that is correlated to the distance given by any global or local PE, such as the distance between two words.*	• Pair-wise node distances from heat kernels, random-walks, Green's function, graph geodesic [3, 34, 41], or any local/global PE. • Gradient of eigenvectors [3, 34] or any local/global PE. • Boolean indicating if two nodes are in the same cluster.
Local SE *node features*	Allow a node to understand what substructures it is a part of. *Given an SE of radius m, the more similar the m-hop subgraphs around two nodes are, the closer their local SE will be.*	• Degree of a node [59]. • Diagonal of the m-steps random-walk matrix [15]. • Time-derivative of the heat-kernel diagonal (gives the degree at $t = 0$). • Enumerate or count predefined structures such as triangles, rings, etc. [6, 64]. • Ricci curvature [51].
Global SE *graph features*	Provide the network with information about the global structure of the graph. *The more similar two graphs are, the closer their global SE will be.*	• Eigenvalues of the Adjacency or Laplacian matrices [34]. • Graph properties: diameter, girth, number of connected components, number of nodes, number of edges, nodes-to-edges ratio.
Relative SE *edge features*	Allow two nodes to understand how much their structures differ. *Edge embedding that is correlated to the difference between any local SE.*	• Gradient of any local SE. • Boolean indicating if two nodes are in the same sub-structure [5] (similar to the gradient of sub-structure enumeration).

Recipes

positional/structural encoding

LapPE,
→ SAN(Spectral Attention Network) extended version, add to the node features of the graph and passed to fully-connected Transformer

RWSE,
→ Using the LSPE(Learnable STructural and Positional Encodings.) random-walk diffusion based positional encoding scheme.

SignNet,
→ general basis symmetries (fresh eigenvector selection)

EquivStableLapPE
→ Separating channel for update the original node features and positional features. and utilize extra positional features 2nd and 3rd smallest eigenvalues and Rotation equivariance fashion.

local message-passing

GatedGCN,
→ gate will close to let the information flow from neighbor j to vertex i, or it will open to stop it.

$$h_i^{\ell+1} = f^{\ell}_{\text{G-GCNN}}\left(h_i^{\ell} , \{h_j^{\ell} : j \to i\} \right) = \text{ReLU}\left(U^{\ell} h_i^{\ell} + \sum_{j \to i} \eta_{ij} \odot V^{\ell} h_j^{\ell}\right)$$

GINE,
→

$$\mathbf{x}'_i = h_{\mathbf{\Theta}}\left((1+\epsilon)\cdot \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \text{ReLU}(\mathbf{x}_j + \mathbf{e}_{j,i})\right)$$

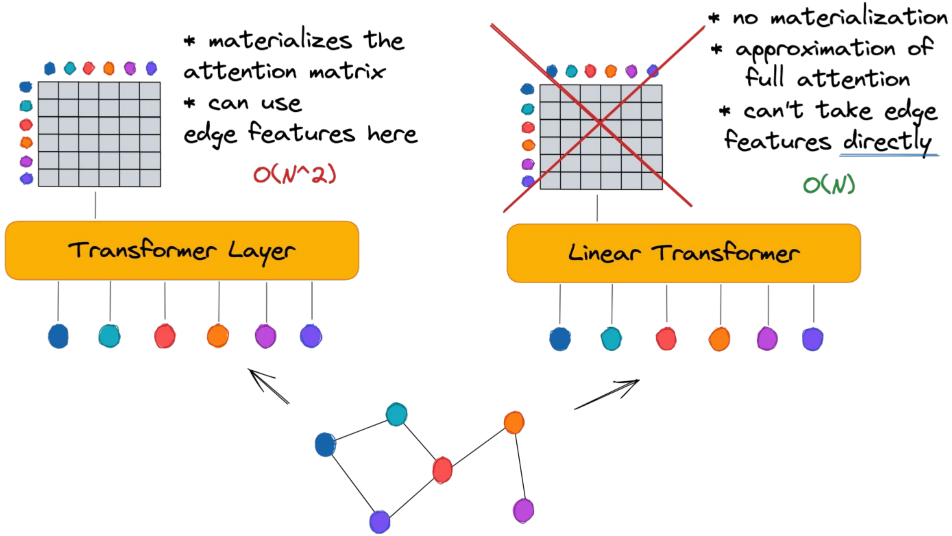
that is able to incorporate edge features $\mathbf{e}_{j,i}$ into the aggregation procedure.

PNA
→ combining multiple aggregators with degree-scalers

global attention

Transformer,
Performer,
BigBird

component 2. Linear Transformer



- interleaving tricks for utilizing edge information at transformer layer

↓

$$h_u^{l+1} = \sum_{v \in \mathcal{N}_u} f(h_u^l, h_v^l, e_{uv}),$$

↓

An example of such function μ_{uv} is the tensor product $\otimes$ of a one-hot encoding unique for each edge o_{uv} and the edge features e_{uv}. For example, if $e_{uv} = [e_1, e_2, e_3]$ and the edge is represented with $o_{uv} = [0, 1, 0, 0]$, then $\mu_{uv} = o_{uv} \otimes e_{uv} = [0, 0, 0, e_1, e_2, e_3, 0, 0, 0, 0, 0, 0]$ satisfies all the above conditions. Although this function requires an exponential increase in the hidden dimension, this is also the case for the Lemma 5 in Xu et al. [57].

↓

To model injective multiset functions for the neighbor aggregation, we develop a theory of "deep multisets", *i.e.*, parameterizing universal multiset functions with neural networks. Our next lemma states that sum aggregators can represent injective, in fact, *universal* functions over multisets.

Lemma 5. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that $h(X) = \sum_{x \in X} f(x)$ is unique for each multiset $X \subset \mathcal{X}$ of bounded size. Moreover, any multiset function g can be decomposed as $g(X) = \phi\left(\sum_{x \in X} f(x)\right)$ for some function ϕ.*

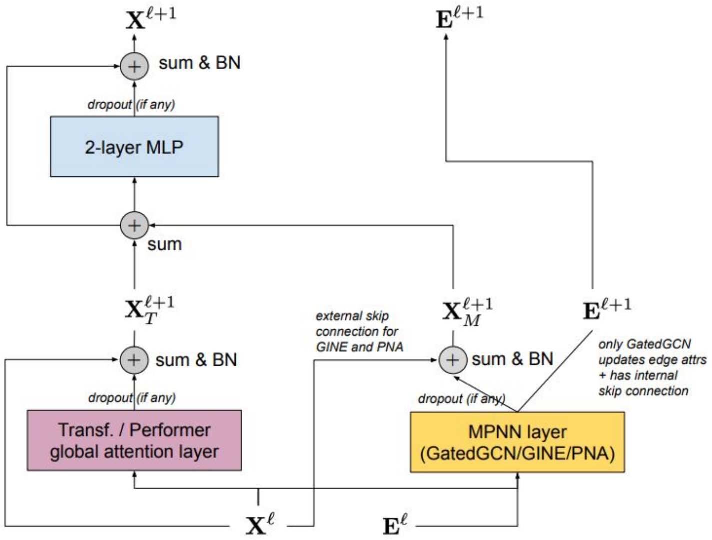
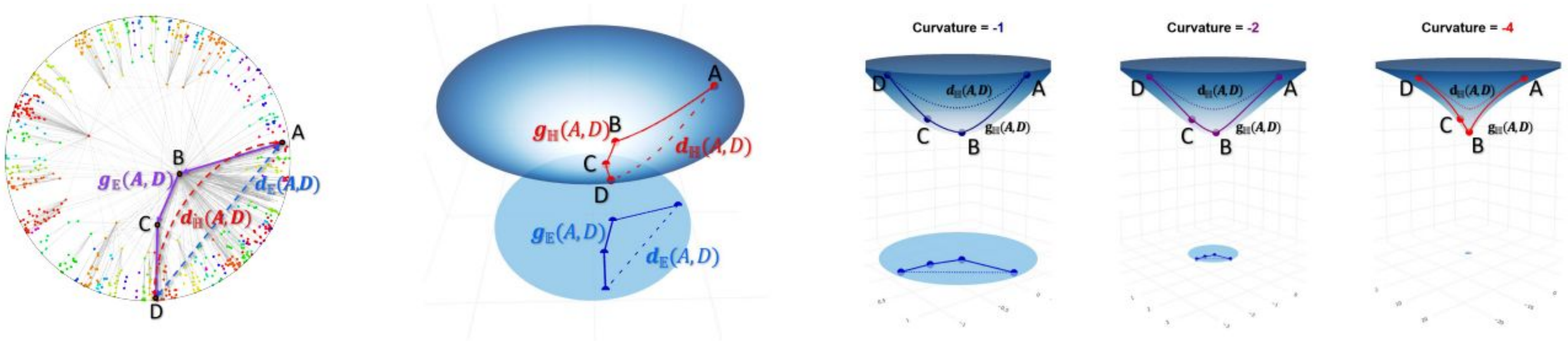


Figure D.1: Modular GPS layer that combines local MPNN and global attention blocks. Local MPNN encodes real edge features into the node-level hidden representations, while global attention mechanism can implicitly make use of this information together with PE/SE to infer relation between two nodes without explicit edge features. After each functional block (an MPNN layer, a global attention layer, an MLP) we apply residual connections followed by batch normalization (BN) [28]. In the 2-layer MLP block we use ReLU activations and its inner hidden dimension is twice the layer-input feature dimensionality d_ℓ. Note, similarly to Transformer, the input and output dimensionality of the GPS-layer as a whole is the same.

$$
\begin{aligned}
\mathbf{X}^{\ell+1}, \mathbf{E}^{\ell+1} &= \mathtt{GPS}^{\ell}\left(\mathbf{X}^{\ell}, \mathbf{E}^{\ell}, \mathbf{A}\right) && (6)\\
\hat{\mathbf{X}}_M^{\ell+1}, \mathbf{E}^{\ell+1} &= \mathtt{MPNN}_e^{\ell}\left(\mathbf{X}^{\ell}, \mathbf{E}^{\ell}, \mathbf{A}\right), && (7)\\
\hat{\mathbf{X}}_T^{\ell+1} &= \mathtt{GlobalAttn}^{\ell}\left(\mathbf{X}^{\ell}\right), && (8)\\
\mathbf{X}_M^{\ell+1} &= \mathtt{BatchNorm}\left(\mathtt{Dropout}\left(\hat{\mathbf{X}}_M^{\ell+1}\right) + \mathbf{X}^{\ell}\right), && (9)\\
\mathbf{X}_T^{\ell+1} &= \mathtt{BatchNorm}\left(\mathtt{Dropout}\left(\hat{\mathbf{X}}_T^{\ell+1}\right) + \mathbf{X}^{\ell}\right), && (10)\\
\mathbf{X}^{\ell+1} &= \mathtt{MLP}^{\ell}\left(\mathbf{X}_M^{\ell+1} + \mathbf{X}_T^{\ell+1}\right) && (11)
\end{aligned}
$$

Open Discussion



(a) Distances in Poincaré disk. (b) Distances projection. (c) Distances with different curvature.

Figure 2. An illustration of different distance metrics in hyperbolic spaces with different curvature. (a) Graph distance (purple solid lines), Euclidean distance (blue dashed line) and hyperbolic geodesics (red dashed curve) on a tree-like graph in Poincaré disk. (b) Graph distance and embedded distance in Poincaré disk (Euclidean projection, blue solid and dashed lines) and hyperboloid (curvature $K = -1$, red solid and dashed curves). (c) Graph distance and hyperbolic distance on the hyperboloid of different curvature.

References & ad

- https://towardsdatascience.com/graphgps-navigating-graph-transformers-c2cc223a051c

- Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., & Beaini, D. (2022). Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv preprint arXiv:2205.12454*.

- Fu, X., Li, J., Wu, J., Sun, Q., Ji, C., Wang, S., ... & Philip, S. Y. (2021, December). ACE-HGNN: Adaptive curvature exploration hyperbolic graph neural network. In *2021 IEEE International Conference on Data Mining (ICDM)* (pp. 111-120). IEEE.

PseudoLab

Next Pseudo Lab study group builder

[Application] Value extraction from real graph data using Network theory & Graph neural network

if u want to upgrade your network analysis & prediction skill , contact me !