Graph Neural Networks with Learnable Structural and Positional Representations

Vijay Prakash Dwivedi
PhD Student, NTU Singapore

with

Anh Tuan Luu (NTU), Thomas Laurent (LMU),
Yoshua Bengio (Mila) and Xavier Bresson (NUS)

LoGaG: Learning on Graphs and Geometry Reading Group

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Introduction Summary

!!
Problem

- Nodes in graphs do not have canonical positional information, like words in sentences. This causes limitations such as the lack of (global) structural information when MP-GNNs learn on graphs.
- As a result, these models exhibit low representation power due to their inability to differentiate simple graph symmetries.

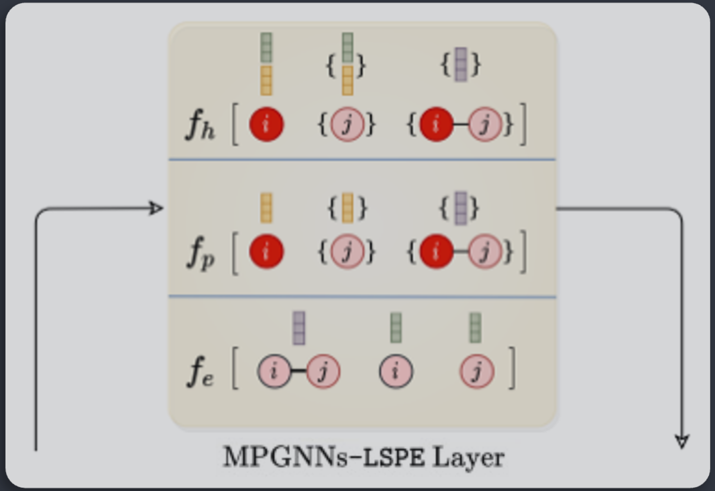
In this work, we consider this topic of graph PEs and propose a framework named LSPE that can be used with any MP-GNNs to **learn positional and structural feature representations** at the same time, thus effectively capturing the two properties and tuning these w.r.t. to the task.

<<
Approach

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

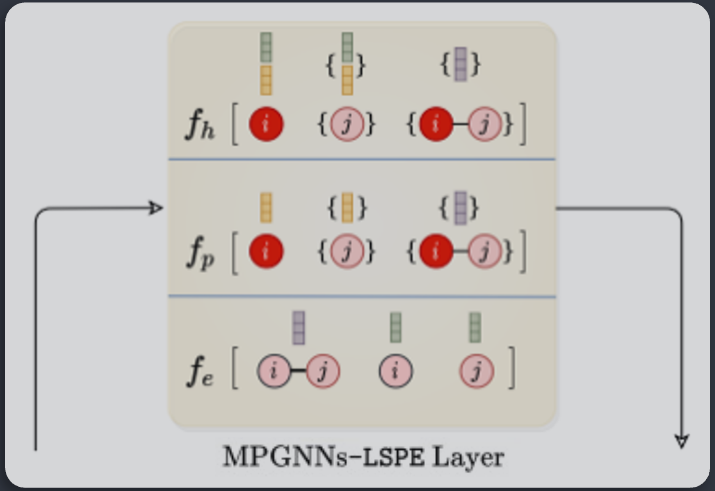
Outline of Presentation

- Motivation
- Background
- Learnable Structural and Positional Encodings (LSPE)
- Numerical Evaluations
- Conclusion



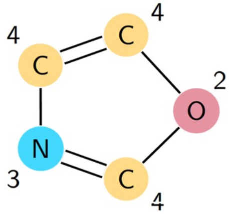
Outline of Presentation

- **Motivation**
- Background
- Learnable Structural and Positional Encodings (LSPE)
- Numerical Evaluations
- Conclusion

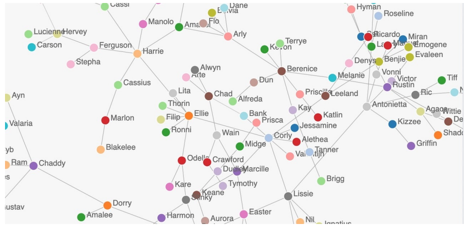


Graph Structured Data

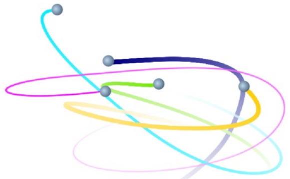
- Graphs are universal language to describe complex systems of bodies and their interactions.



Chemistry
- Learn on molecules and predict chemical properties
- Use in drug repurposing

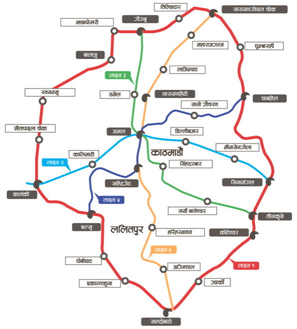


Social networks
- Learn from multi-faceted interactions among users
- Use for commercial and social applications

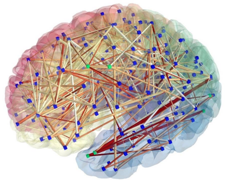


Simple Particles

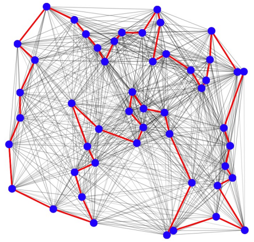
Physics
- Learn from interactions of particles in systems
- Accelerate physics research



Transportation
- Learn from traffic behavior across road networks
- Predict time estimates; efficient transport management



Neuroscience
- Learn functions of brain regions through connectivity
- Accelerate brain-understanding and neuro-disease research



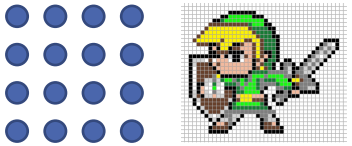
Combinatorial Optimization
- Exploit the fact that most CO problems are rep. as graphs
- Develop better approximated solutions for NP-hard problems

Numerous such examples of graph data and application areas!

Motivation

- Let’s start with some data examples.

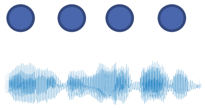
Images



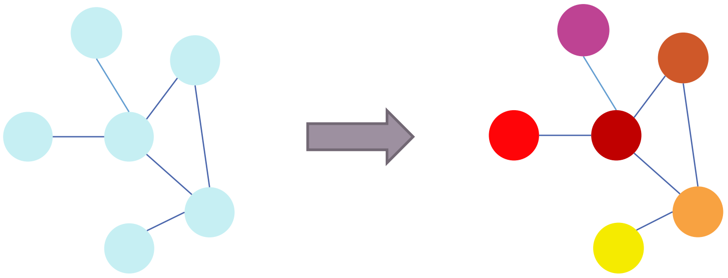
Text

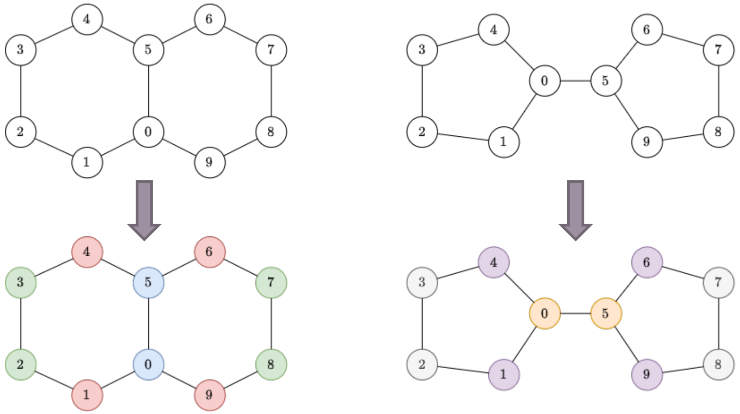
राउर नाँव का ह?

Speech



- CNNs[1] implicitly encode spatial position[2], RNNs build on sequences[3], Transformers use word PE[4].
- What about **graphs** and how GNNs incorporate node positional information?





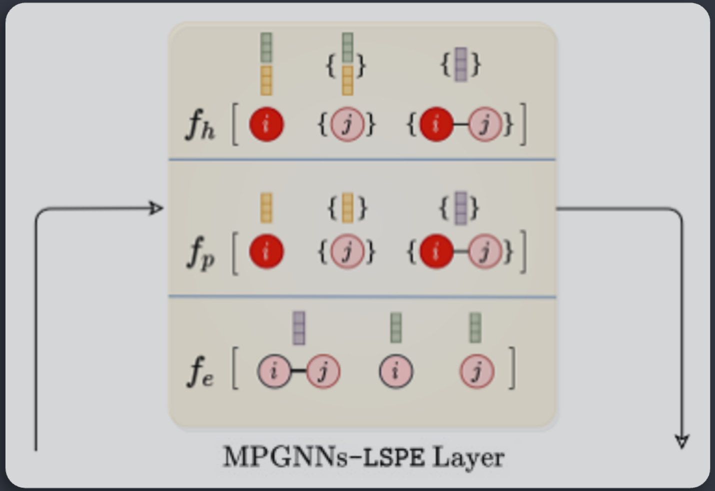
Decalin

Bicyclopentyl

[1] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition.
[2] Islam, M.A., Jia, S. and Bruce, N.D., 2020. How much position information do convolutional neural networks encode?
[3] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory.
[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need.

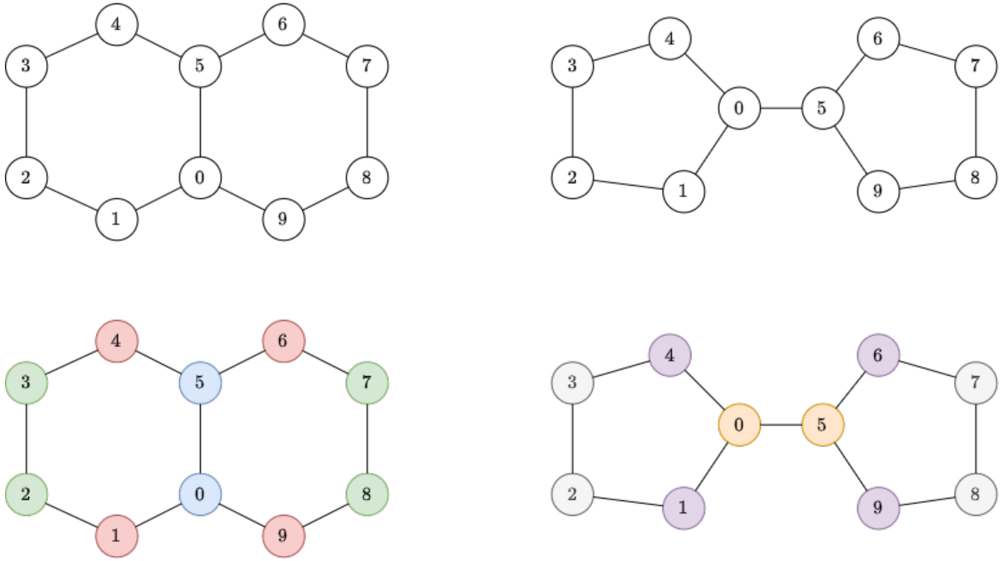
Outline of Presentation

- ❑ Motivation
- ❑ **Background**
- ❑ Learnable Structural and Positional Encodings (LSPE)
- ❑ Numerical Evaluations
- ❑ Conclusion



Background

1. GNN’s theoretical **expressivity**



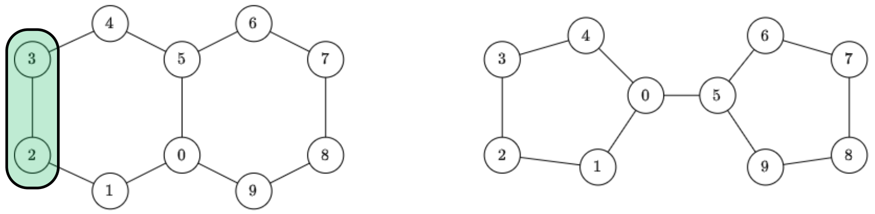
2. Graph **Positional Encoding**

3. **Transformer**-based GNNs

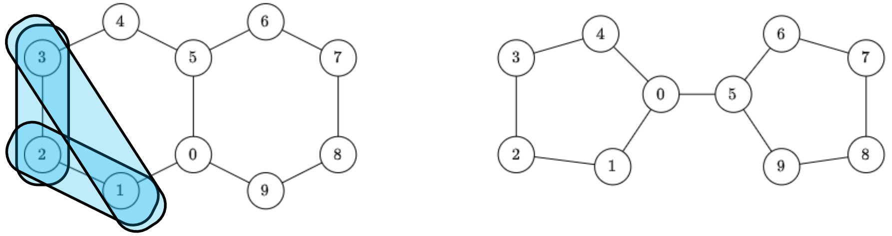
Background

1. GNN's theoretical **expressivity**

- **Message Passing GNNs.**
  -Based on message-passing between nodes [1]
  -Pairwise exchange of information between local neighbors
  -Expressivity bounded by 1-Weisfeiler Leman [2,3]
  -Fail to distinguish simple graphs w.r.t. 1-WL
  -*O(n)* complexity; n: number of nodes



- **Weisfeiler Leman GNNs.**
  -Higher-order GNNs based on the WL test hierarchy (1/2-WL, 3-WL, …, k-WL) [3,4,5,6]
  -Encodes higher-order interactions using k-tuples
  -Can distinguish non-isomorphic graphs w.r.t. k-WL
  -$O(n^2)/O(n^3)$ memory/speed complexity



- Scalable and hybrid WL-GNNs developed recently [6,7]

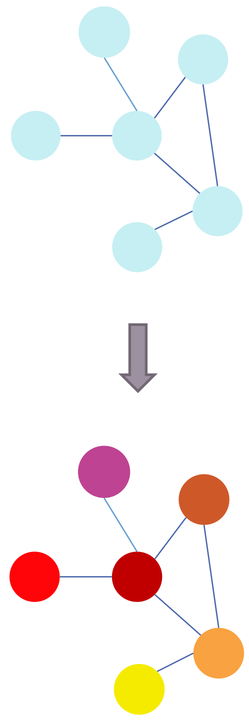
[1] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O. and Dahl, G.E., 2017, July. Neural message passing for quantum chemistry.
[2] Xu, K., Hu, W., Leskovec, J. and Jegelka, S., 2018. How powerful are graph neural networks?
[3] Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G. and Grohe, M., 2019, July. Weisfeiler and leman go neural: Higher-order graph neural networks.
[4] Maron, H., Ben-Hamu, H., Serviansky, H. and Lipman, Y., 2019. Provably powerful graph networks.
[5] Chen, Z., Villar, S., Chen, L. and Bruna, J., 2019. On the equivalence between graph isomorphism testing and function approximation with gnns.
[6] Morris, C., Rattan, G. and Mutzel, P., 2019. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings.
[7] Bodnar, C., Frasca, F., Otter, N., Wang, Y.G., Liò, P., Montufar, G.F. and Bronstein, M., 2021. Weisfeiler and lehman go cellular: Cw networks.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Background

2. Graph **Positional Encoding**

- PE in GNNs help the network disambiguate node symmetries
- MP-GNNs can be more expressive [1] and universal approximators with unique node identifiers [2]



Desired Characteristics:

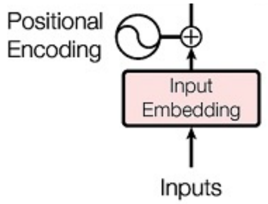
- Unique, Distance-aware, Permutation-equivariant, Efficient/Scalable

Laplacian Positional Encoding [3]

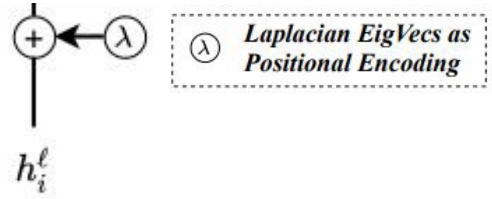
- Based on Laplacian Eigenvectors [4] that embed graphs into a local coordinate system
- Unique and Distance-aware
- Pre-computed from the factorization of the graph Laplacian

$$\Delta = I - D^{-1/2} A D^{-1/2} = U^T \Lambda U$$

- Generalize the PE used in Transformer [5] to graphs



PE in Original Transformer



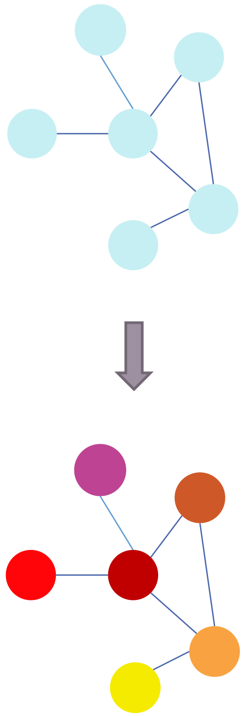
LapPE in GNNs/GraphTransformer

[1] Murphy, R., Srinivasan, B., Rao, V. and Ribeiro, B., 2019, May. Relational pooling for graph representations.
[2] Loukas, A., 2020. What graph neural networks cannot learn: depth vs width.
[3] Dwivedi, V.P., Joshi, C.K., Laurent, T., Bengio, Y. and Bresson, X., 2020. Benchmarking graph neural networks.
[4] Belkin, M. and Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation.
[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need.

Background

2. Graph **Positional Encoding**

Although **LapPE** shows good empirical performance, there are **limitations**:
- Eigenvectors are defined up to ±1
- When selecting k (<<n) eigenvectors, the number of possibilities is 2^k
- During training, the sign of eigenvectors is randomly flipped [1] for uniform sampling among the 2^k possibilities (less than n! of node indexing)

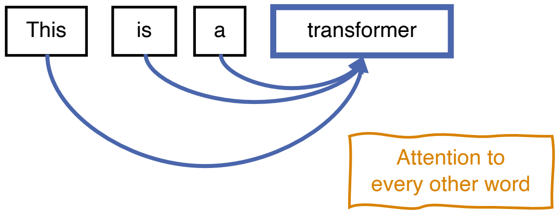


Other PE based works for GNNs:
- Position aware GNNs that use anchor sets and relative distances [2]
- Distance Encoding w.r.t. a node set used at input layer and as aggregation controllers [3]
- Random walk with Restart as topological embeddings [4]
- PEs used in Transformers for Graphs (next slides →)

[1] Dwivedi, V.P., Joshi, C.K., Laurent, T., Bengio, Y. and Bresson, X., 2020. Benchmarking graph neural networks.
[2] Loukas, A., 2020. What graph neural networks cannot learn: depth vs width.
[3] Li, P., Wang, Y., Wang, H. and Leskovec, J., 2020. Distance Encoding--Design Provably More Powerful GNNs for Structural Representation Learning.
[4] Ahmadi, A.H.K., 2020. Memory-based graph networks.

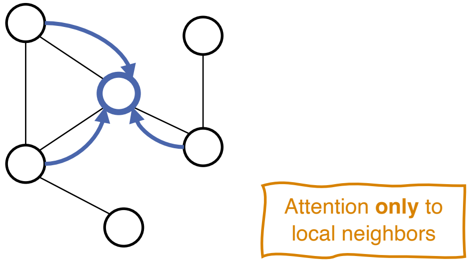
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Background



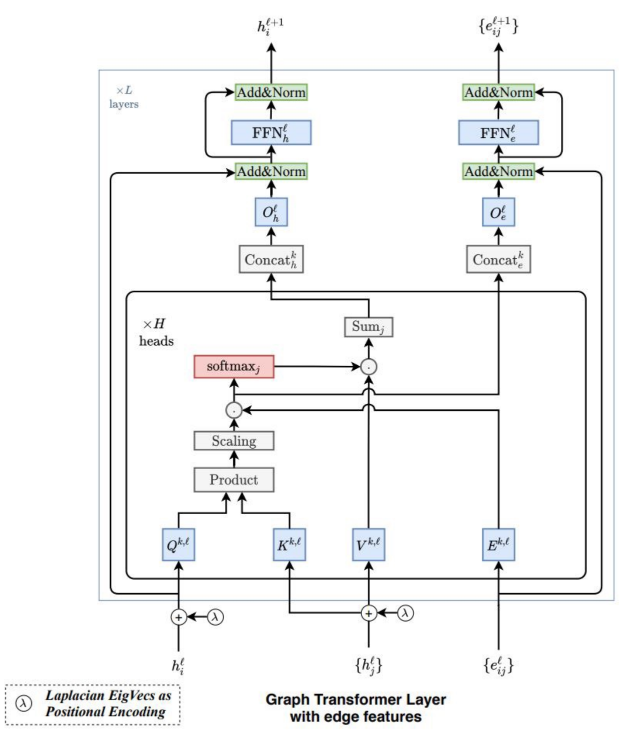
3. Transformer-based GNNs

- Attention based GNNs first proposed in GATs [1]
- Transformers can be viewed as a special case of GNNs on fully connected graph of words [2]
- This **connection** of Transformer and GNNs led to several recent works!



GraphTransformer

- GraphTransformer uses LapPE and local attention to generalize original Transformers to graphs [3]
- However, being similar to MP-GNNs it is susceptible to information bottleneck [4]
- Direct use of full attention without the use of appropriate positional and structural encodings does not work well [3]



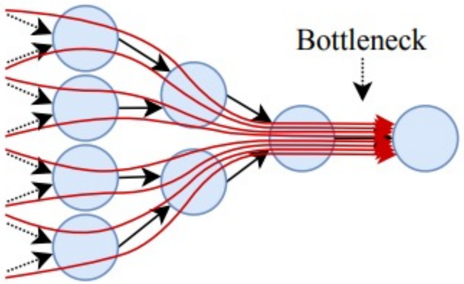


Image from [4]

[1] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y., 2017. Graph attention networks.
[2] Joshi, C., 2020. Transformers are graph neural networks.
[3] Dwivedi, V.P. and Bresson, X., 2020. A generalization of transformer networks to graphs.
[4] Alon, U. and Yahav, E., 2020. On the bottleneck of graph neural networks and its practical implications.

Background

3. Transformer-based GNNs

- Fully connected Transformer-GNNs recently proposed which perform better thanks to **PE-focused** innovations!

Spectral Attention Networks (SAN) [1]

- SAN use a Learnable PE module that applies a Transformer encoder on a sequence of eigenvalues/vectors to generate a fixed sized PE
- During full-attention in the main Transformer, separate learnable parameters are maintained for real and non-real edges

← Injecting PE at node inputs

GraphiT [2]

- Use of diffusion geometry to capture short and long-range information
- The diffusion distance is multiplied with attention score

← Injecting PE with attention score

Graphormer [3]

- Use of centrality, spatial and edge encoding to improve node inputs and attention.

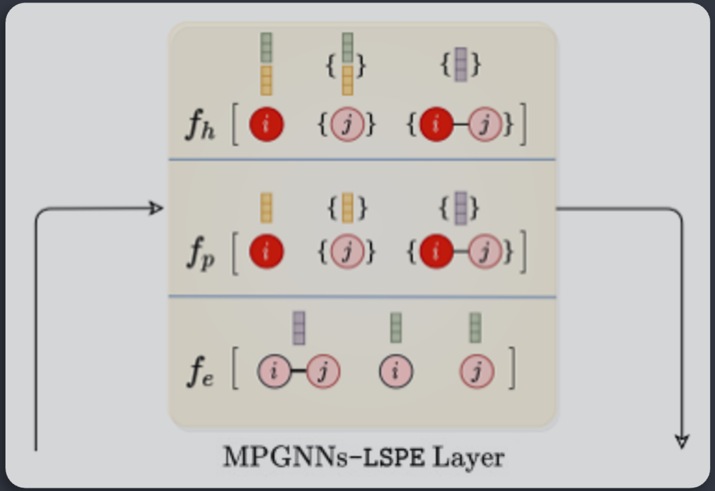
← Injecting PE with node inputs as well as attention score

Note: Other previous works such as Graph-BERT [4] use a combination of structural and relative encodings after a link-less subgraph batching

[1] Kreuzer, D., Beaini, D., Hamilton, W.L., Létourneau, V. and Tossou, P., 2021. Rethinking Graph Transformers with Spectral Attention.
[2] Mialon, G., Chen, D., Selosse, M. and Mairal, J., 2021. GraphiT: Encoding Graph Structure in Transformers.
[3] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y. and Liu, T.Y., 2021. Do Transformers Really Perform Bad for Graph Representation?
[4] Zhang, J., Zhang, H., Xia, C. and Sun, L., 2020. Graph-bert: Only attention is needed for learning graph representations.

Outline of Presentation

- ❑ Motivation
- ❑ Background
- ❑ **Learnable Structural and Positional Encodings (LSPE)**
- ❑ Numerical Evaluations
- ❑ Conclusion



Learnable Structural and Positional Encodings

1. Standard MP-GNNs

$$\text{MP-GNNs}: \quad h_i^{\ell+1} = f_h\left(h_i^{\ell}, \{h_j^{\ell}\}_{j\in\mathcal{N}_i}, e_{ij}^{\ell}\right),\ h_i^{\ell+1}, h_i^{\ell} \in \mathbb{R}^d,$$
$$e_{ij}^{\ell+1} = f_e\left(h_i^{\ell}, h_j^{\ell}, e_{ij}^{\ell}\right),\ e_{ij}^{\ell+1}, e_{ij}^{\ell} \in \mathbb{R}^d,$$

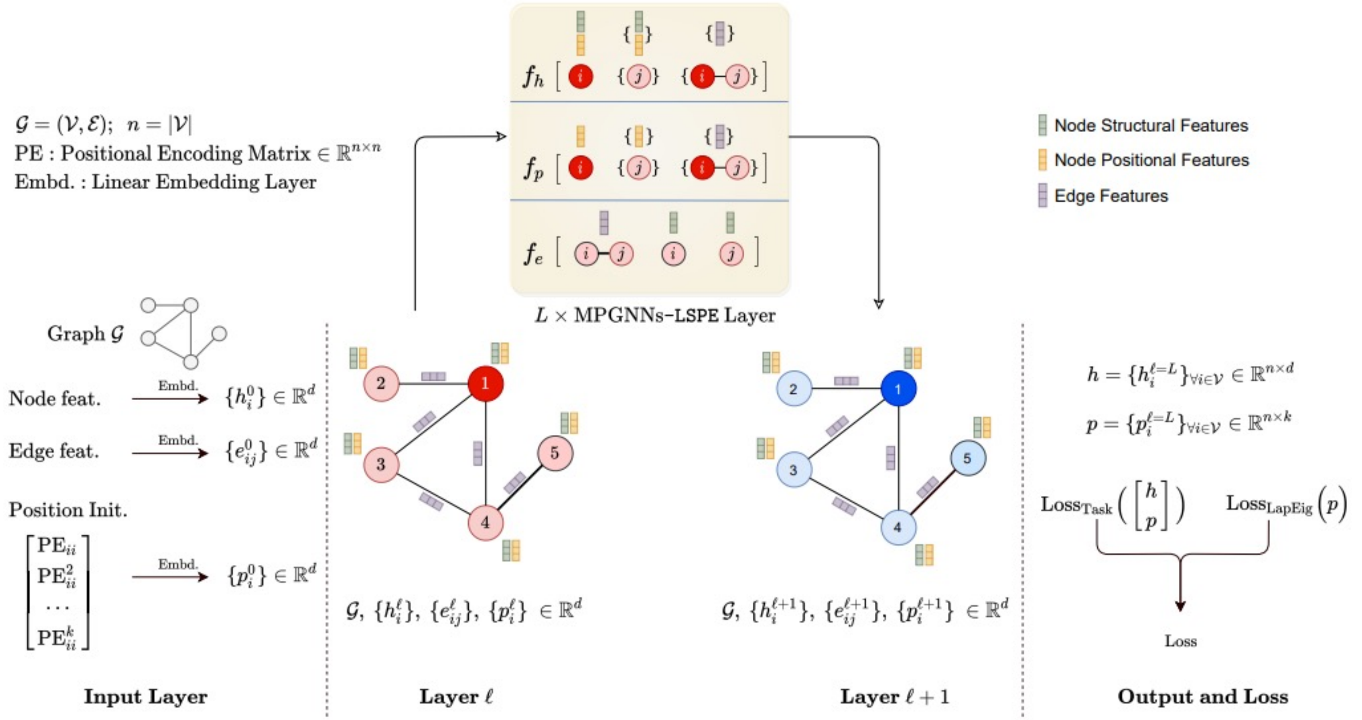
2. MP-GNNs with PE injected with the input node features

$$\text{with initial } h_i^{\ell=0} = \text{LL}_h\left(\begin{bmatrix} h_i^{\text{in}} \\ p_i^{\text{in}} \end{bmatrix}\right) = D^0 \begin{bmatrix} h_i^{\text{in}} \\ p_i^{\text{in}} \end{bmatrix} + d^0 \in \mathbb{R}^d,$$
$$\text{and } e_{ij}^{\ell=0} = \text{LL}_e(e_{ij}^{\text{in}}) = B^0 e_{ij}^{\text{in}} + b^0 \in \mathbb{R}^d,$$

3. **[Proposed] Learning structural and positional representations** at every layer

$$\text{MP-GNNs-LSPE}: \quad h_i^{\ell+1} = f_h\left(\begin{bmatrix} h_i^{\ell} \\ p_i^{\ell} \end{bmatrix}, \left\{\begin{bmatrix} h_j^{\ell} \\ p_j^{\ell} \end{bmatrix}\right\}_{j\in\mathcal{N}_i}, e_{ij}^{\ell}\right),\ h_i^{\ell+1}, h_i^{\ell} \in \mathbb{R}^d,$$
$$e_{ij}^{\ell+1} = f_e\left(h_i^{\ell}, h_j^{\ell}, e_{ij}^{\ell}\right),\ e_{ij}^{\ell+1}, e_{ij}^{\ell} \in \mathbb{R}^d,$$
$$p_i^{\ell+1} = f_p\left(p_i^{\ell}, \{p_j^{\ell}\}_{j\in\mathcal{N}_i}, e_{ij}^{\ell}\right),\ p_i^{\ell+1}, p_i^{\ell} \in \mathbb{R}^d,$$

- We decouple positional and structural representations and learn each separately
- Initial PE to be chosen (next slide →)
- The proposed modification is generic and applies to any model that fits to MP-GNNs
- Additionally, a positional loss can be used for tuning final positional representations



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Learnable Structural and Positional Encodings

Initial PE

- Choice of initial PE to be used in MP-GNNs-LSPE is critical
- If we use LapPE which can provide unique node representation, it brings the limitations as discussed

$$p_i^{\text{LapPE}} = [\ U_{i1}, U_{i2}, \cdots, U_{ik}\ ] \in \mathbb{R}^k$$

- We propose **RWPE** based on the Random Walk diffusion process

$$p_i^{\text{RWPE}} = [\ \text{RW}_{ii}, \text{RW}_{ii}^2, \cdots, \text{RW}_{ii}^k\ ] \in \mathbb{R}^k$$

- We use RW_{ii} from every k^{th} step to construct RWPE of node i
- Thus, RWPE **encodes** the landing probability of a node to itself in 1 to k steps of random walk
- RWPE closely based on Distance Encoding (DE) [1] but we do not consider all relative RW_{ij} as in DE

Meaningful higher-order structure information!

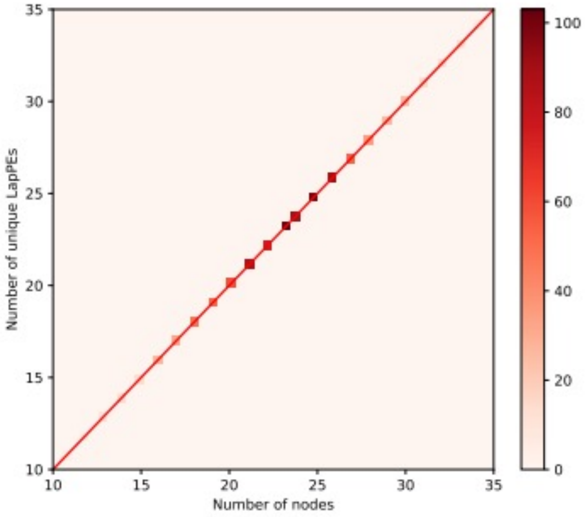
[1] Li, P., Wang, Y., Wang, H. and Leskovec, J., 2020. Distance Encoding--Design Provably More Powerful GNNs for Structural Representation Learning.

Learnable Structural and Positional Encodings

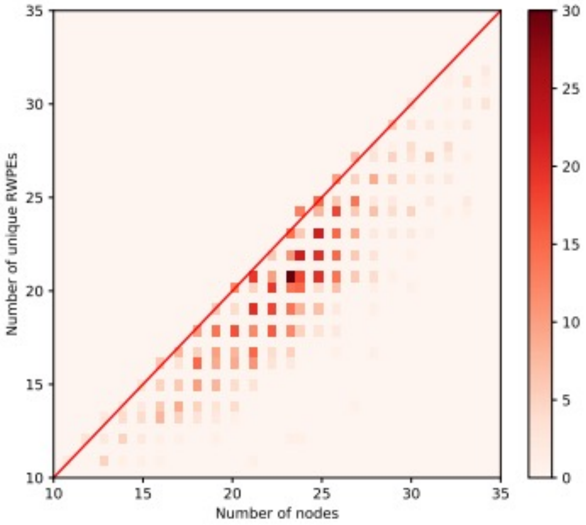
Initial PE: RWPE

Characteristics of RWPE

- Unique representation for a node “given a distinct k-hop” neighborhood when considering a sufficient k

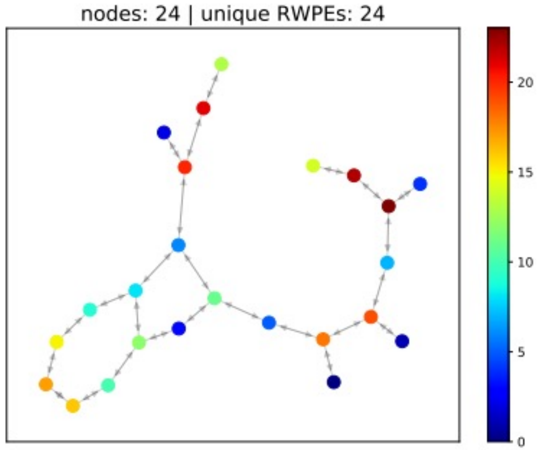


(a) LapPE, $k = 36$

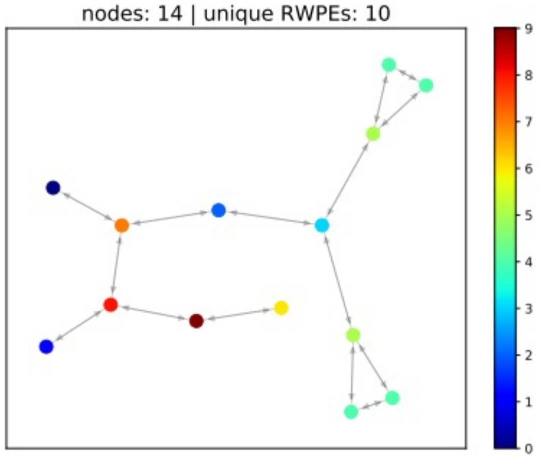


(b) RWPE, $k = 24$

ZINC valset: number of nodes vs. number of unique PE
the point intensity is the number of graphs



(a) ZINC molecule (val index 91)



(b) ZINC molecule (val index 212)

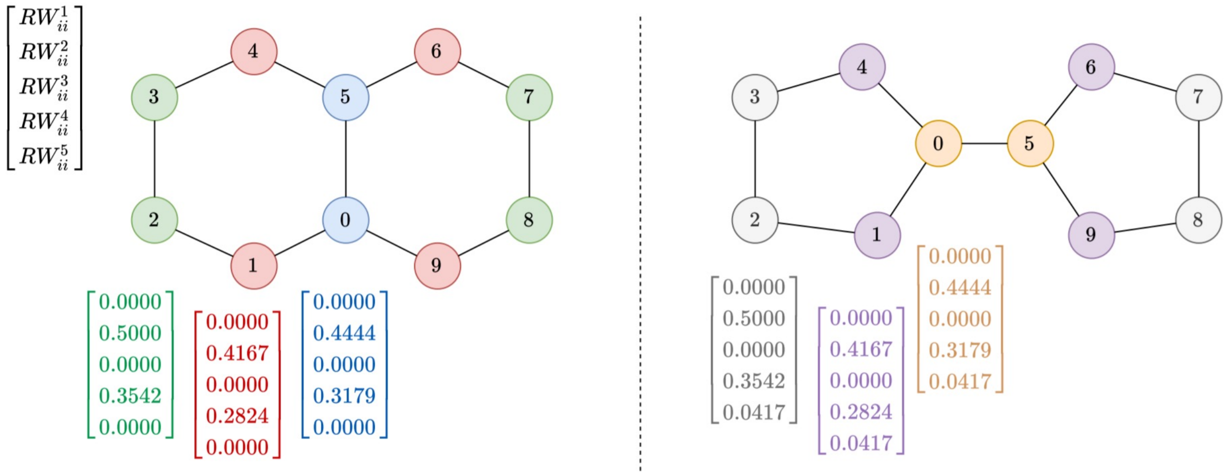
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Learnable Structural and Positional Encodings

Initial PE: RWPE

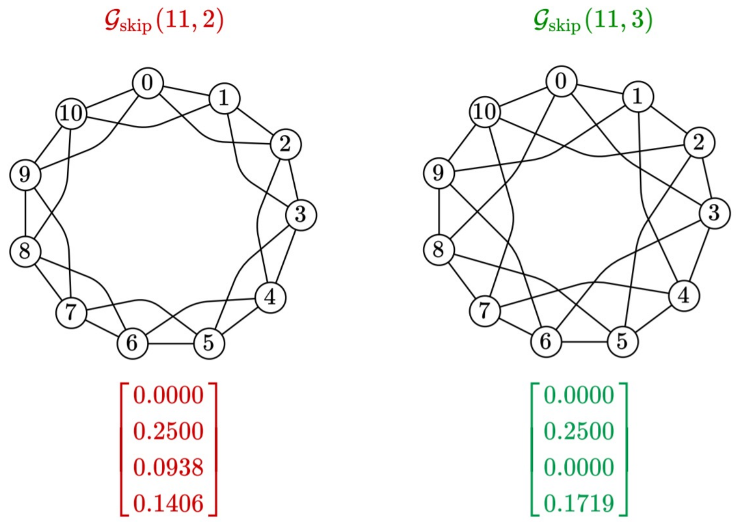
Characteristics of RWPE

- No sign ambiguity as in LapPE
- Can distinguish graph-pairs that failed to be classified correctly by MP-GNNs (1-WL)



Decalin

Bicyclopentyl



A pair of Circular Skip Link Graphs

Learnable Structural and Positional Encodings

A. Instantiation of LSPE with sparse MP-GNNs

$$\text{MP-GNNs-LSPE}: \quad h_i^{\ell+1} = f_h\left(\begin{bmatrix} h_i^\ell \\ p_i^\ell \end{bmatrix}, \left\{\begin{bmatrix} h_j^\ell \\ p_j^\ell \end{bmatrix}\right\}_{j\in\mathcal{N}_i}, e_{ij}^\ell\right),\ h_i^{\ell+1}, h_i^\ell \in \mathbb{R}^d,$$
$$e_{ij}^{\ell+1} = f_e\left(h_i^\ell, h_j^\ell, e_{ij}^\ell\right),\ e_{ij}^{\ell+1}, e_{ij}^\ell \in \mathbb{R}^d,$$
$$p_i^{\ell+1} = f_p\left(p_i^\ell, \{p_j^\ell\}_{j\in\mathcal{N}_i}, e_{ij}^\ell\right),\ p_i^{\ell+1}, p_i^\ell \in \mathbb{R}^d,$$

1. GatedGCN [1] -LSPE

$$h^{\ell+1}, e^{\ell+1}, p^{\ell+1} = \text{GatedGCN-LSPE}\left(h^\ell, e^\ell, p^\ell\right),\ h \in \mathbb{R}^{n\times d}, e \in \mathbb{R}^{E\times d}, p \in \mathbb{R}^{n\times d},$$
$$\text{with}\ \ h_i^{\ell+1} = h_i^\ell + \text{ReLU}\left(\text{BN}\left(A_1^\ell \begin{bmatrix} h_i^\ell \\ p_i^\ell \end{bmatrix} + \sum_{j\in\mathcal{N}(i)} \eta_{ij}^\ell \odot A_2^\ell \begin{bmatrix} h_j^\ell \\ p_j^\ell \end{bmatrix}\right)\right),$$
$$e_{ij}^{\ell+1} = e_{ij}^\ell + \text{ReLU}\left(\text{BN}\left(\hat{\eta}_{ij}^\ell\right)\right),$$
$$p_i^{\ell+1} = p_i^\ell + \tanh\left(C_1^\ell p_i^\ell + \sum_{j\in\mathcal{N}(i)} \eta_{ij}^\ell \odot C_2^\ell p_j^\ell\right),$$
$$\text{and}\ \ \eta_{ij}^\ell = \frac{\sigma\left(\hat{\eta}_{ij}^\ell\right)}{\sum_{j'\in\mathcal{N}(i)} \sigma\left(\hat{\eta}_{ij'}^\ell\right) + \epsilon},$$
$$\hat{\eta}_{ij}^\ell = B_1^\ell h_i^\ell + B_2^\ell h_j^\ell + B_3^\ell e_{ij}^\ell,$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[1] Bresson, X. and Laurent, T., 2017. Residual gated graph convnets.

Learnable Structural and Positional Encodings

A. Instantiation of LSPE with sparse MP-GNNs

$$\text{MP-GNNs-LSPE}: \quad h_i^{\ell+1} = f_h\left(\begin{bmatrix} h_i^\ell \\ p_i^\ell \end{bmatrix}, \left\{\begin{bmatrix} h_j^\ell \\ p_j^\ell \end{bmatrix}\right\}_{j\in\mathcal{N}_i}, e_{ij}^\ell\right), \; h_i^{\ell+1}, h_i^\ell \in \mathbb{R}^d,$$

$$e_{ij}^{\ell+1} = f_e\left(h_i^\ell, h_j^\ell, e_{ij}^\ell\right), \; e_{ij}^{\ell+1}, e_{ij}^\ell \in \mathbb{R}^d,$$

$$p_i^{\ell+1} = f_p\left(p_i^\ell, \{p_j^\ell\}_{j\in\mathcal{N}_i}, e_{ij}^\ell\right), \; p_i^{\ell+1}, p_i^\ell \in \mathbb{R}^d,$$

2. PNA [1] -LSPE

$$h^{\ell+1}, p^{\ell+1} = \text{PNA-LSPE}\left(h^\ell, e^0, p^\ell\right), \; h \in \mathbb{R}^{n\times d}, e^0 \in \mathbb{R}^{E\times d}, p \in \mathbb{R}^{n\times d},$$

$$\text{with } h_i^{\ell+1} = h_i^\ell + \text{LReLU}\left(\text{BN}\left(U_h^\ell\left(\begin{bmatrix} h_i^\ell \\ p_i^\ell \end{bmatrix}, \bigoplus_{j\in\mathcal{N}(i)} M_h^\ell\left(\begin{bmatrix} h_i^\ell \\ p_i^\ell \end{bmatrix}, e_{ij}^0, \begin{bmatrix} h_j^\ell \\ p_j^\ell \end{bmatrix}\right)\right)\right)\right),$$

$$p_i^{\ell+1} = p_i^\ell + \tanh\left(U_p^\ell\left(p_i^\ell, \bigoplus_{j\in\mathcal{N}(i)} M_p^\ell\left(p_i^\ell, e_{ij}^0, p_j^\ell\right)\right)\right),$$

$$\text{and } \bigoplus = \begin{bmatrix} I \\ S(D, \alpha=1) \\ S(D, \alpha=-1) \end{bmatrix} \otimes \begin{bmatrix} \mu \\ \sigma \\ \max \\ \min \end{bmatrix},$$

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[1] Corso, G., Cavalleri, L., Beaini, D., Liò, P. and Veličković, P., 2020. Principal neighbourhood aggregation for graph nets.

Learnable Structural and Positional Encodings

B. Instantiation of LSPE with Transformer-GNNs*

1. SAN [1] -LSPE

$$h^{\ell+1}, p^{\ell+1} = \text{SAN-LSPE}\left(h^\ell, e^0, p^\ell\right),\ h \in \mathbb{R}^{n\times d}, e^0 \in \mathbb{R}^{n\times n\times d}, p \in \mathbb{R}^{n\times d},$$

$$\text{with}\ h_i^{\ell+1} = \text{BN}\left(\bar{h}_i^{\ell+1} + W_2^\ell\, \text{ReLU}\left(W_1^\ell\, \bar{h}_i^{\ell+1}\right)\right) \in \mathbb{R}^d$$

$$\bar{h}_i^{\ell+1} = \text{BN}\left(h_i^\ell + \hat{h}_i^{\ell+1}\right) \in \mathbb{R}^d,$$

$$\hat{h}_i^{\ell+1} = O^\ell\left(\Big\Vert_{k=1}^{H} \sum_{j\in\mathcal{V}} \frac{w_{ij}^{k,\ell}}{\sum_{j'\in\mathcal{V}} w_{ij'}^{k,\ell}}\, v_j^{k,\ell}\right) \in \mathbb{R}^d,$$

$$w_{ij}^{k,\ell} = \begin{cases} \frac{1}{1+\gamma}\cdot \exp(A_{ij}^{k,\ell}) & \text{if } ij \in E \\ \frac{\gamma}{1+\gamma}\cdot \exp(\bar{A}_{ij}^{k,\ell}) & \text{if } ij \notin E \end{cases},$$

$$\begin{cases} A_{ij}^{k,\ell} = q_i^{k,\ell^T} \text{diag}(c_{ij}^{k,\ell}) k_j^{k,\ell} / \sqrt{d_k} \in \mathbb{R} & \text{if } ij \in E \\ \bar{A}_{ij}^{k,\ell} = \bar{q}_i^{k,\ell^T} \text{diag}(\bar{c}_{ij}^{k,\ell}) \bar{k}_j^{k,\ell} / \sqrt{d_k} \in \mathbb{R} & \text{if } ij \notin E \end{cases}$$

$$Q^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} W_Q^{k,\ell},\ K^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} W_K^{k,\ell},\ V^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} W_V^{k,\ell} \in \mathbb{R}^{n\times d_k}$$

$$\bar{Q}^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} \bar{W}_Q^{k,\ell},\ \bar{K}^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} \bar{W}_K^{k,\ell},\ \bar{V}^{k,\ell} = \begin{bmatrix} h^\ell \\ p^\ell \end{bmatrix} \bar{W}_V^{k,\ell} \in \mathbb{R}^{n\times d_k}$$

$$C^{k,0} = e^0 W_e^k,\ \bar{C}^{k,0} = e^0 \bar{W}_e^k \in \mathbb{R}^{n\times n\times d_k}$$

$$\text{and}\ p_i^{\ell+1} = p_i^\ell + \tanh\left(O_p^\ell\left(\Big\Vert_{k=1}^{H} \sum_{j\in\mathcal{V}} \frac{w_{p,ij}^{k,\ell}}{\sum_{j'\in\mathcal{V}} w_{p,ij'}^{k,\ell}}\, v_{p,j}^{k,\ell}\right)\right) \in \mathbb{R}^d,$$

2. GraphiT [2] –LSPE

Here we use SAN based architecture for separate parameters for real and non-real edges and modify the weight-score using GraphiT as:

$$w_{ij}^{k,\ell} = \begin{cases} K_{ij}\cdot \exp(A_{ij}^{k,\ell}) & \text{if } ij \in E \\ K_{ij}\cdot \exp(\bar{A}_{ij}^{k,\ell}) & \text{if } ij \notin E \end{cases}$$

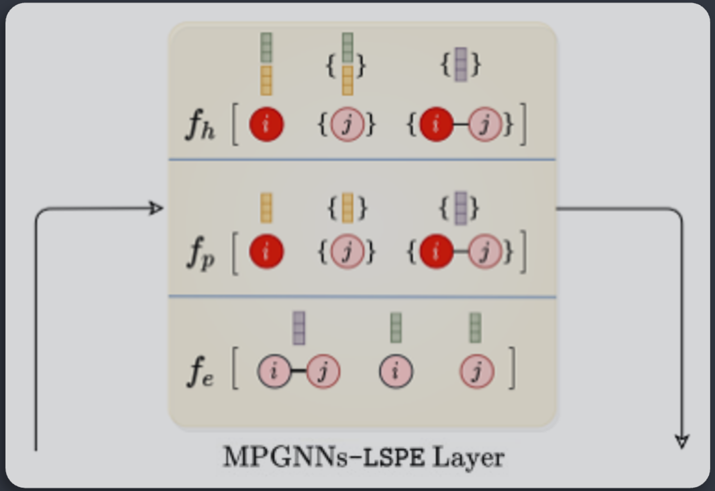
NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

[1] Kreuzer, D., Beaini, D., Hamilton, W.L., Létourneau, V. and Tossou, P., 2021. Rethinking Graph Transformers with Spectral Attention.
[2] Mialon, G., Chen, D., Selosse, M. and Mairal, J., 2021. GraphiT: Encoding Graph Structure in Transformers.

Footnote: * with modifications

Outline of Presentation

- ❑ Motivation
- ❑ Background
- ❑ Learnable Structural and Positional Encodings (LSPE)
- ❑ **Numerical Evaluations**
- ❑ Conclusion



Numerical Evaluations

Step by step empirical evaluation towards the final GNN-LSPE architecture

Instance: GatedGCN-LSPE **Dataset:** ZINC **Metric:** MAE (lower is better)

Insights:

- Adding PE improves performance against not using one
- RWPE outperforms LapPE while simply injecting these at the input
- Use of PE at the input layer as opposed with final learnt representations

- Simply adding RWPE leads to overfit
- Using LSPE alleviates this overfit and improves performance significantly

	Model	Init PE	LSPE	Final h	L	#Param	Test MAE±s.d.	Train MAE±s.d.	#Epochs	Epoch/Total
1	GatedGCN	x	x	h^L	16	504309	0.251±0.009	0.025±0.005	440.25	8.76s/1.08hr
2	GatedGCN	LapPE	x	h^L	16	505011	0.202±0.006	0.033±0.003	426.00	8.91s/1.22hr
3	GatedGCN	RWPE	x	h^L	16	505947	0.122±0.003	0.013±0.003	436.25	9.14s/1.28hr
4	GatedGCN	x	x	$[h^L, \textbf{RWPE}]$	16	515249	0.249±0.012	0.024±0.002	437.50	10.05s/1.55hr
5	GatedGCN	LapPE	✓	h^L	16	516722	0.202±0.008	0.032±0.005	405.25	15.10s/1.84hr
6	GatedGCN	LapPE	✓	$[h^L, p^L]$	16	520734	0.196±0.008	0.023±0.004	454.00	15.22s/2.06hr
7	GatedGCN	RWPE	✓	h^L	16	518150	0.100±0.006	0.018±0.012	395.00	15.09s/1.73hr
8	GatedGCN	RWPE	✓	$[h^L, p^L]$	16	522870	0.093±0.003	0.014±0.003	440.75	15.17s/1.99hr

- **At last**, using the final layer positional and structural features for the task with LSPE gives the best performance

Numerical Evaluations

All experiments on 3 molecular datasets

Insights:

- LSPE consistently enhances the capabilities of existing GNNs on the 4 instances evaluated
- On these datasets, sparse GNNs show better performance compared to fully-connected Transformer GNNs
- Positional Loss improves test score on ZINC slightly while in general it leads to overfit on training data

Dataset	Model	Init PE	LSPE	PosLoss	L	#Param	TestMAE±s.d.	TrainMAE±s.d.	Epochs	Epoch/Total
ZINC	GatedGCN	x	x	x	16	504309	0.251±0.009	0.025±0.005	440.25	8.76s/1.08hr
ZINC	GatedGCN	**LapPE**	x	x	16	505011	0.202±0.006	0.033±0.003	426.00	8.91s/1.22hr
ZINC	GatedGCN	**RWPE**	✓	x	16	522870	0.093±0.003	0.014±0.003	440.75	15.17s/1.99hr
ZINC	GatedGCN	**RWPE**	✓	✓	16	522870	**0.090±0.001**	0.013±0.004	460.50	33.06s/4.39hr
ZINC	PNA	x	x	x	16	369235	0.141±0.004	0.020±0.003	451.25	79.67s/10.03hr
ZINC	PNA	**RWPE**	✓	x	16	503061	**0.101±0.003**	0.026±0.012	443.75	126.06s/15.77hr
ZINC	SAN	x	x	x	10	501314	0.181±0.004	0.017±0.004	433.50	74.33s/9.23hr
ZINC	SAN	**RWPE**	✓	x	10	588066	**0.104±0.004**	0.016±0.002	462.50	134.74s/17.23hr
ZINC	GraphiT	x	x	x	10	501313	0.181±0.006	0.021±0.003	493.25	63.54s/9.37hr
ZINC	GraphiT	**RWPE**	✓	x	10	588065	**0.106±0.002**	0.028±0.002	420.50	125.39s/14.84hr
	Model	**Init PE**	**LSPE**	**PosLoss**	L	**#Param**	**TestAUC±s.d.**	**TrainAUC±s.d.**	**Epochs**	**Epoch/Total**
MOLTOX21	GatedGCN	x	x	x	8	1003739	0.772±0.006	0.933±0.010	304.25	5.12s/0.46hr
MOLTOX21	GatedGCN	**LapPE**	x	x	8	1004355	0.774±0.007	0.921±0.006	275.50	5.23s/0.48hr
MOLTOX21	GatedGCN	**RWPE**	✓	x	8	1063821	**0.775±0.003**	0.906±0.003	246.50	5.99s/0.63hr
MOLTOX21	PNA	x	x	x	8	5244849	0.755±0.008	0.876±0.014	214.75	6.25s/0.38hr
MOLTOX21	PNA	**RWPE**	✓	x	8	5357393	**0.781±0.013**	0.901±0.013	249.75	9.87s/0.73hr
MOLTOX21	PNA	**RWPE**	✓	✓	8	5357393	0.778±0.006	0.906±0.012	245.00	24.09s/1.70hr
MOLTOX21	SAN	x	x	x	10	957871	**0.744±0.007**	0.915±0.015	279.75	18.06s/1.44hr
MOLTOX21	SAN	**RWPE**	✓	x	10	1051017	0.744±0.008	0.918±0.018	281.75	30.82s/2.84hr
MOLTOX21	GraphiT	x	x	x	10	957870	0.743±0.003	0.919±0.023	276.50	16.73s/1.36hr
MOLTOX21	GraphiT	**RWPE**	✓	x	10	1051788	**0.746±0.010**	0.934±0.016	279.75	27.92s/2.57hr
	Model	**Init PE**	**LSPE**	**PosLoss**	L	**#Param**	**TestAP±s.d.**	**TrainAP±s.d.**	**Epochs**	**Epoch/Total**
MOLPCBA	GatedGCN	x	x	x	8	1008263	0.262±0.001	0.401±0.057	190.50	149.10s/7.91hr
MOLPCBA	GatedGCN	**LapPE**	x	x	8	1008879	0.266±0.002	0.391±0.003	177.00	152.94s/8.29hr
MOLPCBA	GatedGCN	**RWPE**	✓	x	8	1068721	**0.267±0.002**	0.403±0.006	181.00	206.43s/11.64hr
MOLPCBA	PNA	x	x	x	4	6550839	0.279±0.003	0.448±0.004	129.25	174.75s/6.34hr
MOLPCBA	PNA	**RWPE**	✓	x	4	6521029	**0.287±0.003**	0.392±0.002	334.50	202.59s/23.68hr

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Numerical Evaluations

Comparison with baselines and state-of-the-art GNNs

(a) ZINC

Model	Test MAE
GCN	0.367±0.011
GAT	0.384±0.007
GatedGCN-LapPE	0.202±0.006
GT	0.226±0.014
SAN	0.139±0.006
Graphormer	0.122±0.006
GatedGCN-LSPE	**0.090±0.001**

(b) OGBG-MOLTOX21

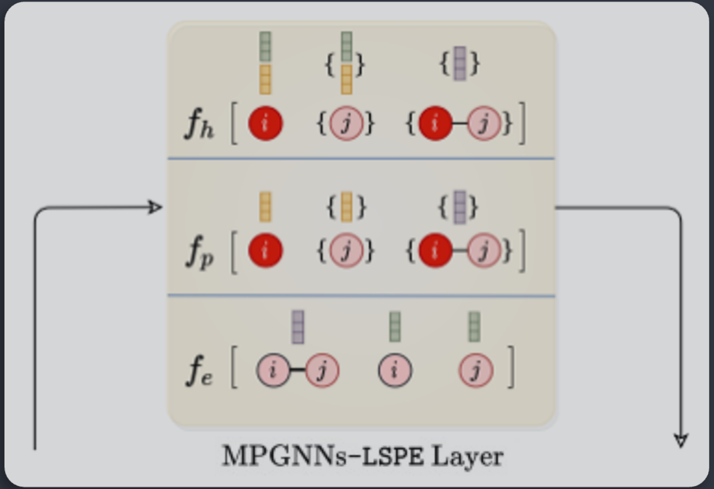
Model	Test ROC-AUC
GCN	0.7529±0.0069
GCN-VN	0.7746±0.0086
GIN	0.7491±0.0051
GIN-VN	0.7757±0.0062
GatedGCN-LapPE	0.7743±0.0073
PNA-LSPE	**0.7808±0.0130**

(c) OGBG-MOLPCBA

Model	Test AP
GIN	0.2266±0.0028
GIN-VN	0.2703±0.0023
DeeperGCN-VN	0.2781±0.0038
PNA	0.2838±0.0035
DGN	0.2885±0.0030
PHC-GNN	**0.2947±0.0026**
PNA-LSPE	0.2873±0.0027

Outline of Presentation

- ❑ Motivation
- ❑ Background
- ❑ Learnable Structural and Positional Encodings (LSPE)
- ❑ Numerical Evaluations
- ❑ **Conclusion**



Conclusion

- Decoupling and learning structural and positional representations at every layer improves existing GNNs – both sparse and fully-connected.
- The initialization of PE is critical and contributes greatly towards improving a GNN.
- LSPE helps the feature representations to be tuned for the task and improves generalization performance.
- Compared to fully-connected Transformer-GNNs, the sparse GNNs perform better on the molecular datasets that we considered for evaluation and are efficient at the same time.
- The proposed LSPE architecture is a general framework that can be applied to improve any GNN that fits in the message-passing framework.

Questions?

Thank you!

Vijay Prakash Dwivedi
https://vijaydwivedi.com.np
VIJAYPRA001@e.ntu.edu.sg

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE