Extended Abstract Track

Range-aware Positional Encoding via High-order Pretraining: Theory and Practice

Editors: List of editors' names

Abstract

Based on Wavelet Positional Encoding of Ngo et al. (2023), we propose **HOPE-WavePE** (**H**igh-**O**rder **P**ermutation **E**quivariant **Wave**let **P**ositional **E**ncoding) a novel pre-training strategy for positional encoding that is equivariant under the permutation group and is sensitive to the length and diameter of graphs downstream tasks. Since our approach relies solely on the graph structure, it is domain-agnostic and adaptable to datasets from various domains, therefore paving the wave for developing general graph structure encoders and graph foundation models. We theoretically demonstrate that such equivariant pretraining schema can approximate the training target for abitrarily small tolerance. We also evaluate HOPE-WavePE on graph-level prediction tasks of different areas and show its superiority compared to other methods. We release our source code upon the acceptance.

Keywords: Topological learning, Permutation Equivariance, Positional Encoding, Graph Neural Network

1. Introduction

One of the fastest-growing areas in machine learning is graph representation learning, with impactful applications in biomedicine, molecular chemistry, and network science. Most graph neural networks (GNNs) rely on the message-passing framework that processes graph-structured data by exchanging the vectorized information between nodes on graphs along their edges. Albeit achieving remarkable results in a wide range of tasks on graph data, message-passing neural networks (MPNNs) possess several fundamental limits, including expressiveness Xu et al. (2019), over-smoothing Chen et al. (2020), and over-squashing Topping et al. (2022). In recent years, transformer-based architectures Yun et al. (2019); Kreuzer et al. (2021); Dwivedi and Bresson (2020); Ying et al. (2021) have emerged as powerful alternatives to address the mentioned issues of MPNN. However, graph transformers (GTs) and VN-augmented MPNNs disregard the underlying structure of graph data by altering inherent connections among the nodes (i.e., shortening all paths to two). This disregard may explain their limitations in several graph-level prediction tasks. To address this, positional and structural encodings (PSE) are commonly used to enhace structural information in modern GNNs.

Our work builds on similar approaches to Wang et al. (2022) and Liu et al. (2023), where we pre-train an encoder to capture node positional information in an unsupervised setting. However, unlike previous works, our encoder is high-order equivariant, enabling it to capture multi-scale properties of graph structures using Wavelet positional encodings Ngo et al. (2023). These learned positional features can be adapted to various downstream tasks and generalize well to domain-specific datasets.. Furthermore, realizing the limitations of previous pretraining schema by relying on domain features You et al. (2021); Xu et al. (2021); Zhu et al. (2021); You et al. (2020); Jiao et al. (2020), we aim to generalize the

learnability by exploiting the intrinsic structure of graphs, i.e. adjacency matrices. Finally, unlike structural encodings like random walk whose receptive field is limited to the hop length, our encoding method aims to be sensitive to the graph size while still capturing long-range structural information.

In this work, we propose a new pretraining approach on graphs that leverages the reconstruction of graph structures from the Wavelet signals to generalize structural information on graph data, thus enabling transfer learning to various downstream tasks across various domains of different ranges. Our contributions are three-fold as follows:

- We propose a high-order structural pretrained models for graph-structured data and a loss-masking technique that leverages high-order interactions of nodes on graphs while being aware of the graph size and diameter.
- We theoretically prove that pretraining by reconstruction with multi-resolution Wavelet signals can make autoencoder learn node state after an arbitrarily walk of length d, which can contain both local and global information of graph structures.
- We empirically show that such pretraining scheme can enhance the performance of supervised models when fine-tuned on downstream datasets of different domains, indicating the generalizabiilty and effectiveness of pretrained structural encoding compared with other domain-specific pretraining methods.

2. Methodology

2.1. Spectral Graph Wavelet Tensors

Given the eigendecomposition of the normalized Laplacian $\widetilde{L} = U\Lambda U^T$, where Λ is the diagonal matrix of eigenvalues. The graph Wavelet transforms construct a set of spectral graph Wavelet as bases to project the graph signal from the vertex domain to the spectral domain as:

$$\psi_s = U\Lambda_s U^T, \tag{1}$$

here, $\Lambda_s = \mathrm{diag}(g(s\lambda_1), \cdots, g(s\lambda_n))$ is the scaling matrix of the eigenvalues. The scaling function g takes the eigenvalue λ_i and an additional scaling factor s as inputs, indicating how a signal diffuses away from node i at scale s; we select $g_s(\lambda) = \exp(-s\lambda)$ as the heat kernel. This means that we can vary the scaling parameter s to adjust the neighborhoods surrounding a center node. For computation efficiency, we use the fast graph wavelet transform Hammond et al. (2011) to approximate these wavelets with complexity $O(|E| \times M)$, where M is the order of polynomials and E is the set of edges.

2.2. Constructing Long-range Pretraining on Domain-Agnostic Data

Encoder Given a second-order wavelet tensor $\mathbf{W} \in \mathbb{R}^{n\times n\times k}$, the encoder $\mathcal{E}$ encodes $\mathbf{W}$ into a latent matrix $\mathbf{Z} = \mathcal{E}(\mathbf{W}) \in \mathbb{R}^{n\times d_\ell}$, the encoder $\mathcal{E}$ can be composed of many equivariant operators. Furthermore, to reduce redundancy caused by high-order training, we extract only two equivariant mappings for the encoder, diagonal and row sum.

Decoder The decoder $\mathcal{D}$ lifts $\mathbf{Z}$ back to a high-order feature map $\mathcal{F} = \mathcal{D}(\mathbf{Z}) \in \mathbb{R}^{n\times n\times d}$. Here, We use the outer product and diagonal operator, which represent structural and positional informationn respectively. Afterward, we use a channel-wise multi-layer perceptron (MLP) $\phi : \mathbb{R}^{n\times n\times d} \mapsto \mathbb{R}^{n\times n\times r}$ to map $\mathcal{F}$ to a concatenated high-degree adjacency matrix in binary values. Specifically, let $\mathbf{A}_j$ be the binary matrix of j-hop neighbor in a graph and $\widehat{\mathbf{A}}_j$ be its prediction. The final MLP network returns the predicted array

$$\phi(\mathbf{Z}) = \begin{bmatrix} \widehat{\mathbf{A}}_{s_1} & \widehat{\mathbf{A}}_{s_2} & \dots & \widehat{\mathbf{A}}_{s_r} \end{bmatrix}, \tag{2}$$

where $s_1, s_2, \dots, s_r$ are natural degrees to be chosen. In this work, we let these values follow a exponential pattern, which highlights the range-diversity.

Theoretically, we show that with sufficient budget, our pretraining schema can reach abitraily high accuracy, the full proof is provided at Appendix 8.

Theorem 1 *For any $\epsilon > 0$ and real coefficients $\theta_1, \theta_2, \dots, \theta_d$, there exists a high-order autoencoder network $\varphi : \mathbb{R}^{n\times n\times d} \to \mathbb{R}^{n\times n}$ such that*

$$\left\| \varphi(\mathbf{Z}) - \sum_{j=1}^{r} \theta_j \mathbf{A}_j \right\| < \epsilon.$$

In order to build balanced learning in edges and non-edges in each hop length, we delve into the pair-level connection of each $\mathbf{A}_s$ in the aforementioned adjacency array. In particular, we mask out random edges and non-edges in the reconstruction loss such that they are of equal quantity:

$$\mathcal{L}_{\mathbf{M}}(\mathbf{A}_{s_i}, \widehat{\mathbf{A}}_{s_i}) = \text{BinCrossEntropy}\left(\mathbf{M}_i \odot \mathbf{A}_{s_i}, \mathbf{M}_i \odot \widehat{\mathbf{A}}_{s_i}\right),$$

where $\odot$ is the elementwise matrix product. This masking technique can filter out redundant adjacencies of graphs. Specifically, $\mathbf{M}_i$ filters out $\mathbf{A}_{s_i}$ if the entries are all ones and are dependent on the graph structure. By this way, it filters out unnecessary hop lengths for shorter graphs, preserving the generalizability of these hops for longer graphs. This lets us combine datasets of different graph sizes and only extract meaningful relevant features.

3. Experiments

Pretraining We pretrained a high-order autoencoder on MolPCBA Wu et al. (2018) and Peptides-func Dwivedi et al. (2022). During this pretraining stage, the autoencoder focuses on learning a set of topological hops, represented by the concatenated tensor $\{\mathbf{A}_{s_i}\}_{i=1}^{r}$. By excluding chemical features during pretraining, we granted the network versatility, enabling it to adapt to downstream tasks that use different feature representations. To incorporate multi-scale information, we constructed a 4-channel Wavelet tensor for each graph sample, with scaling factors of $[1, 2, 4, 16]$. The autoencoder architecture consisted of an encoder and a decoder, each containing a three layer MLP. The decoded output undergoes another three-layer MLP of dimensionality $[16, 32, 64, 32, 16]$ is used to learn the adjacency tensor $\{\mathbf{A}_{s_i}\}_{i=1}^{r}$. We divide the MolPCBA dataset into a train-valid ratio of 9:1 and use the prepaired train-valid set for Peptides-func.

3.1. Results

Since we pretrained our HOPE-WavePE autoencoder on the MolPCA and Peptides-func, We focus on evaluating the transferrability of HOPE-WavePE to some out-of-distribution (OOD) datasets. This should highlight the independent of HOPE-WavePE in terms of domain and indicates the learning continuous spectrum range of our method, motivated by the continuously decaying nature of the Wavelet transform on graphs. We evaluate HOPE-WavePE on six diverse datasets from the TUDataset benchmark Morris et al. (2020): a small molecule dataset (MUTAG), two chemical compound datasets (NCI1 and NCI109), a macromolecule dataset (PROTEINS) and two social network datasets (IMDB-B and IMDB-M). The results in 1 demonstrate that GIN augmented with HOPE-WavePE significantly outperforms other complicated high-order networks like IGN Maron et al. (2019b), CIN, and PPGNs on three out of six datasets.

Table 1: Experimental results on TU datasets. The methods are evaluated by Accuracy % (↑). The reported results are means and standard deviations of runnings over five random seeds. Top 3 results are highlighted, including **First**, **Second**, and **Third**.

Method	MUTAG	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M
RWK Gärtner et al. (2003)	79.2 ± 2.1	59.6 ± 0.1	>3 days	-	-	-
GK ($k = 3$) Shervashidze et al. (2009)	81.4 ± 1.7	71.4 ± 0.3	62.5 ± 0.3	62.4 ± 0.3	-	-
PK Neumann et al. (2014)	76.0 ± 2.7	73.7 ± 0.7	82.5 ± 0.5	-	-	-
WL kernel Shervashidze et al. (2011)	90.4 ± 5.7	75.0 ± 3.1	**86.0** ± 1.8	-	73.8 ± 3.9	50.9 ± 3.8
DCNN Atwood and Towsley (2016)	-	61.3 ± 1.6	56.6 ± 1.0	-	49.1 ± 1.4	33.5 ± 1.4
DGCNN Zhang et al. (2018)	85.8 ± 1.8	75.5 ± 0.9	74.4 ± 0.5	-	70.0 ± 0.9	47.8 ± 0.9
IGN Maron et al. (2019b)	83.9 ± 13.0	76.6 ± 5.5	74.3 ± 2.7	72.8 ± 1.5	72.0 ± 5.5	48.7 ± 3.4
GIN Xu et al. (2019)	89.4 ± 5.6	76.2 ± 2.8	82.7 ± 1.7	-	75.1 ± 5.1	52.3 ± 2.8
PPGNs Maron et al. (2019a)	90.6 ± 8.7	**77.2** ± 4.7	83.2 ± 1.1	**82.2** ± 1.4	73.0 ± 5.8	50.5 ± 3.6
Natural GN de Haan et al. (2020)	89.4 ± 1.6	71.7 ± 1.0	82.4 ± 1.3	-	73.5 ± 2.0	51.3 ± 1.5
GSN Bouritsas et al. (2023)	**92.2** ± 7.5	76.6 ± 5.0	83.5 ± 2.0	-	**77.8** ± 3.3	**54.3** ± 3.3
SIN Bodnar et al. (2021)	-	76.4 ± 3.3	82.7 ± 2.1	-	**75.6** ± 3.2	**52.4** ± 2.9
CIN Bodnar et al. (2022)	**92.7** ± 6.1	**77.0** ± 4.3	**83.6** ± 1.4	**84.0** ± 1.6	**75.6** ± 3.7	**52.7** ± 3.1
GIN + HOPE-WavePE (ours)	**93.6** ± 5.8	**79.5** ± 4.81	**84.5** ± 2.0	**84.1** ± 1.9	**76.0** ± 3.7	**52.7** ± 2.9

4. Conclusion

We have introduced HOPE-WavePE, a novel high-order permutation-equivariant pretraining method specifically designed for graph-structured data. Our approach leverages the inherent connectivity of graphs, eliminating reliance on domain-specific features. This enables HOPE-WavePE to generalize effectively across diverse graph types and domains. The superiority of HOPE-WavePE is demonstrably proven through both theoretical and empirical analysis. Finally, we have discussed the potential of HOPE-WavePE as a foundation for a general graph structural encoder. A promising future direction will be to focus on optimizing the scalability of this approach.

References

James Atwood and Don Towsley. Diffusion-convolutional neural networks, 2016.

Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montúfar, Pietro Liò, and Michael Bronstein. Weisfeiler and lehman go topological: Message passing simplicial networks, 2021.

Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montúfar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks, 2022.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023. doi: 10.1109/TPAMI.2022.3154319.

Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between mpnn and graph transformer. *arXiv preprint arXiv:2301.11956*, 2023.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3438–3445, Apr. 2020. doi: 10.1609/aaai.v34i04.5747. URL https://ojs.aaai.org/index.php/AAAI/article/view/5747.

Pim de Haan, Taco Cohen, and Max Welling. Natural graph networks, 2020.

Michaël Defferrard, Lionel Martin, Rodrigo Pena, and Nathanaël Perraudin. Pygsp: Graph signal processing in python. *URL https://github. com/epfl-lts2/pygsp*, 2017.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *CoRR*, abs/2012.09699, 2020. URL https://arxiv.org/abs/2012.09699.

Vijay Prakash Dwivedi, Ladislav Rampasek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=in7XC5RcjEn.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 129–143, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyong Zhu. Subgraph contrast for scalable self-supervised graph representation learning. In *2020 IEEE international conference on data mining (ICDM)*, pages 222–231. IEEE, 2020.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21618–21629. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b4fd1d2cb085390fbbadae65e07876a7-Paper.pdf.

Renming Liu, Semih Cantürk, Olivier Lapointe-Gagné, Vincent Létourneau, Guy Wolf, Dominique Beaini, and Ladislav Rampášek. Graph positional and structural encoder. *arXiv preprint arXiv:2307.07107*, 2023.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/bb04af0f7ecaee4aae62035497da1387-Paper.pdf.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=Syx72jC9tm.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Marion Neumann, R. Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102:209 – 245, 2014. URL https://api.semanticscholar.org/CorpusID:14487732.

Nhat Khang Ngo, Truong Son Hy, and Risi Kondor. Multiresolution graph transformers and wavelet positional encoding for learning long-range and hierarchical structures. *The Journal of Chemical Physics*, 159(3):034109, 07 2023. ISSN 0021-9606. doi: 10.1063/5.0152833. URL https://doi.org/10.1063/5.0152833.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv:2205.12454*, 2022.

Espen Sande, Carla Manni, and Hendrik Speleers. Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis. *Numerische Mathematik*, 144(4):889–929, January 2020. ISSN 0945-3245. doi: 10.1007/s00211-019-01097-9. URL http://dx.doi.org/10.1007/s00211-019-01097-9.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In David van Dyk and Max Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 488–495, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/shervashidze09a.html.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561, 2011. URL http://jmlr.org/papers/v12/shervashidze11a.html.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1548–1554. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/214. URL https://doi.org/10.24963/ijcai.2021/214. Main Track.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=e95i1IHcWj.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11548–11558. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xu21g.html.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OeWooOxFwDa.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/3fe230348e9a12c13120749e3f9fa4cd-Paper.pdf.

Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12121–12132. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/you21a.html.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/9d63484abb477c97640154d40595a3bb-Paper.pdf.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*, 2021.

Appendix A. First Appendix

This is the first appendix.

Appendix B. Theoretical analysis

Notations. Throughout this section, for $\mathbf{X} \in \mathbb{R}^{m\times n}$, we denote $\mathbf{X}[i:j]$ as the indiced matrix of X from row i to row j and $X[i]$ as the i-th row of $\mathbf{X}$.

B.1. Preliminary results

The approximation methodology we use is the spline approximation on the eigenvalues of the laplacians, which later . First we need to define a scalar k-spline function on a bounded domain $[a,b] \subset \mathbb{R}$. Let

$$a =: \eta_1 < \eta_2 < \cdots < \eta_{N+1} := b$$

such that $\eta_{j+1} = \eta_j + (b-a)/N$, these points are called uniform knots in the interval $[a,b]$. Assume that $\mathcal{P}_k$ is the class of polynomial up degree at most k.

Definition 2 *A scalar function f is called a k-spline function in the uniformly-divided interval $[a,b]$ of $N+1$ knots if the follows conditions are satisfied:*

- $f(x) = p_i(x) \qquad \forall\, x \in [\eta_i, \eta_{i+1}],\ i = \overline{1,N}$ *for some* $p_i \in \mathcal{P}_k$,
- *Derivative of f up to degree k is continuous.*

For convenience, we denote the class of all such functions in Definition 2 as $\mathcal{S}_{k,N}^{[a,b]}$.

Lemma 3 *(spline approximation power) Sande et al. (2020) Given a scalar function u of smoothness order $k+1$ in the range $[a,b]$ divided in uniform knots of length h, there exists $p \in \mathcal{S}_{k,N}^{[a,b]}$ such that*

$$\|u - p\| \le \left(\frac{h}{\pi}\right)^k \left\|u^{(k+1)}\right\|$$

where $u^{(k+1)}$ is the $(k+1)$-th order derivative of u.

Our proof strategy incorporate the approximation of high degree polymonial with k-spline methodology. Since both wavelet and random walk second feature embeddings' largest eigenvalue is 1 and smallest eigenvalue is -1, we can let $a = -1$ and $b = 1$ in our case. Note that in a two-layer MLP, the number of subintervals $[\eta_i, \eta_{i+1}]$ should correspond to the width in the hidden layer, which is equavalent to $1/h$. Therefore, Lemma 3 yields the condition in which a tolerance ϵ is satisfied.

Letting $u(x) = x^d$ for some $d > k$, then u is obviously smooth up to arbitrary order, thus it is also smooth of order k. We will let $\|\cdot\|$ be the max norm in the interval $[0,1]$, then we have

$$\left\|u^{(k+1)}\right\|^{1/k} = (d(d-1)\dots(d-k))^{1/k} \sup_{x\in[0,1]} x^{\frac{d-k+1}{k}} \le d^{1+\frac{1}{k}}$$

Combine this with the statement of Lemma 3 we deduce that the error defined by the max norm will be less than ϵ if

$$w := \frac{1}{h} \ge \pi^{-1}\epsilon^{-\frac{1}{k}} d^{1+\frac{1}{k}} = O\left(\epsilon^{-\frac{1}{k}} d^{1+\frac{1}{k}}\right).$$

From here, we deduce an important lemma:

Lemma 4 *Given two natural numbers d and k such that $d > k$, then there exists a two-layer MLP $f : \mathbb{R}^k \to \mathbb{R}$ of hidden width $O\left(\epsilon^{-\frac{1}{k}} d^{1+\frac{1}{k}}\right)$ such that*

$$\left| x^d - f\left(x, x^2, \ldots, x^k\right) \right| < \epsilon$$

Lemma 5 *(First order extension) For any $\epsilon > 0$ and a given natural number $d > k$, there exists a two-layer $\mathbb{S}_n$-equivariant linear* MLP $: \mathbb{R}^{n\times k} \to \mathbb{R}^{n\times r}$ *network with width $O\left(n^{\frac{1}{k}} \epsilon^{-\frac{1}{k}} r^{1+\frac{1}{k}}\right)$ such that*

$$\left\| \mathrm{MLP}\left(\mathbf{E}_k^{(1)}\right) - \mathbf{E}_r^{(1)} \right\| \leq \epsilon.$$

Proof

Assume that Λ^i is the diagonal matrix containing all eigenvalues of ψ_s^i. Applying Lemma 4, we simply see that Λ^q for $q = \overline{k+1, r}$ can be estimated using a two-layer MLP of width $O\left(\epsilon^{-\frac{1}{k}} r^{1+\frac{1}{k}}\right)$ with the max norm error less than ϵ. Formally, let $\widehat{\psi_s^q}$ be the estimation of ψ_s^q and e_i be the error at the i-th entry along the diagonal, then we have that

$$\left\| \widehat{\psi_s^q} - \psi_s^q \right\| = \left\| \sum_{i=1}^{n} e_i \mathbf{u}_i \mathbf{u}_i^\top \right\| \leq \sum_{i=1}^{n} |e_i| \left\| \mathbf{u}_i \mathbf{u}_i^\top \right\| \leq n\epsilon$$

Replacing ϵ with ϵ/n yields the desired result. ■

With enough first-order informations, i.e. sufficiently large r in Lemma 5, we can reconstruct the second-order features up to arbitrarily high degree.

Theorem 6 *(First to second order) Assume that* $\mathrm{rank}(\psi_s - \mathbf{I}_n) \leq r$ *and let $h : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times n\times r}$ be the resolution-wise outer product, then there exists a broadcasted linear feed forward layer $g : \mathbb{R}^r \to \mathbb{R}^d$ such that $(h \circ g)\left(\mathbf{E}_d^{(1)}\right) = \mathbf{E}_r^{(2)}$.*

Proof The first order features are aggregated through an outer product operator and return r square matrices of order n. However, these matrices are all rank one matrices and cannot represent the initial second order features. Since the rank of a square matrix is equivalent to its length minus the multiplicity of the eigenvalue zero, we can see that

$$\mathrm{rank}(\psi_s - \mathbf{I}_n) = \mathrm{rank}(\psi_s^2 - \mathbf{I}_n) = \cdots = \mathrm{rank}(\psi_s^d - \mathbf{I}_n) \leq r$$

Therefore, for all $i = \overline{1, d}$, the matrix $\psi_s^i - \mathbf{I}_n$ can be written as a weighted sum of r rank one matrices produced from the outer product. This concludes the proof.

■

B.2. Proof of Theorem 1

Theorem 7 *For any $\epsilon > 0$ and real coefficients $\theta_0, \theta_1, \ldots, \theta_d$ assume that $\operatorname{rank}(\psi_s - \mathbf{I}_n) \leq r$, then there exists an $\mathbb{S}_n$-equivariant AE $f : \mathbb{R}^{n\times n\times k} \to \mathbb{R}^{n\times n\times r}$ of width $O(n^{1/k} r^{1+1/k} \epsilon^{-1/k})$ and a broadcasted feed forward network $g : \mathbb{R}^r \to \mathbb{R}$ such that*

$$\|(g \circ f)(\mathbf{E}_k) - \mathbf{p}(\psi_s)\| < \epsilon$$

where $\mathbf{p}(\psi_s) = \sum_{j=0}^{d} \theta_j \psi_s^j$.

Proof Encoder. The input tensor is of size $n \times n \times k$, representing second order feature in k different resolutions. The encoder simply operate resolution-wise and take the row-sum through each square matrix. This encoder will output a first order tensor of size $n \times k$. This layer is evidently $\mathbb{S}_n$-equivariant.

Latent. For the latent space, i.e. first-order feature space, we apply Lemma 5 to extend from k resolutions to r resolutions using a two-layer MLP of width $O\left(n^{\frac{1}{k}} \epsilon^{-\frac{1}{k}} r^{1+\frac{1}{k}}\right)$. And since this MLP is also built upon the broadcasting along the n-axis, it is also $\mathbb{S}_n$-equivariant.

Decoder. Applying the content of Theorem 6 we can conclude the proof.

■

After proving the reconstructability of these wavelets, we show how this translate to long-range structure on graphs via the main objective of Theorem 1.

Theorem 8 *For any $\epsilon > 0$ and real coefficients $\theta_1, \theta_2, \ldots, \theta_d$, there exists a two-layer ReLU feed forward network $\varphi : \mathbb{R}^{n\times n\times d} \to \mathbb{R}^{n\times n}$ of hidden dimension $d_h = 2$ such that*

$$\left\| \varphi(\mathbf{E}_d) - \sum_{j=1}^{d} \theta_j \mathbf{A}_j \right\| < \epsilon.$$

Proof For this proof, we need to consider wavelet and random walk separatedly. **Wavelet.** Let $\psi_s = U \Lambda_s U^\top$ where $\Lambda_s = \operatorname{diag}(\exp(-s\lambda_1), \exp(-s\lambda_2), \ldots, \exp(-s\lambda_n))$. We first need to perform a transform on the vector basis $\mathbf{E}_d^{(2)}$. Essentially, the transformations are done independent of the eigenvectors U. Formally, we observe that

$$\begin{pmatrix} e^{-s\lambda_i} - 1 \\ e^{-2s\lambda_i} - 1 \\ \vdots \\ e^{-ds\lambda_i} - 1 \end{pmatrix} \approx \mathbf{A} \begin{pmatrix} \lambda_i - 1 \\ (\lambda_i - 1)^2 \\ \vdots \\ (\lambda_i - 1)^d \end{pmatrix} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ contains the corresponding Chebyshev polynomial expanding coefficients. Note that (3) is essentially the discrete fourier transform, thus the inversed version is simply

$$\begin{pmatrix} \lambda_i - 1 \\ (\lambda_i - 1)^2 \\ \vdots \\ (\lambda_i - 1)^d \end{pmatrix} \approx \mathbf{A}^{-1} \begin{pmatrix} e^{-s\lambda_i} - 1 \\ e^{-2s\lambda_i} - 1 \\ \vdots \\ e^{-ds\lambda_i} - 1 \end{pmatrix}. \tag{4}$$

This means that the power of $\tilde{L}$ up to d can be retrieved via a broadcasted linear layer. Now let ζ be the scalar step function, meaning $\zeta(x) = 1$ for $x > 0$ and 0 otherwise. Then, let φ be a continuous piece-wise linear function such that:

$$\varphi(x) = \begin{cases} 0 & \text{if } x < 0 \\ x/\varepsilon & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

Since this function is a three-piece linear function, it can be represented as a ReLU-based feed forward network with hidden dimension two. And evidently,

$$\|\varphi - \zeta\| \to 0 \quad \text{as} \quad \varepsilon \to 0.$$

Furthermore, it yields that $\zeta\left((\mathbf{I}_n - \tilde{L})^k\right) = \mathbf{A}_k$ for all k. Therefore, we concluded the proof. ■

Appendix C. Additional implementation details

C.1. Datasets

Table 2 presents details of all benchmarking datasets used in our experiments. We focus on improving model performance in graph-level prediction tasks. All datasets contain over 1000 samples, with the average number of nodes per dataset ranging from 13 to over 100.

C.2. Hyperparameter Settings

Pretraining 3 depicts the hyperparameters of our high-order autoencoder and training settings. In general, we used three layers of IGN Maron et al. (2019b) to build the encoder hidden dimensions of $[8, 16, 32]$. The decoder is a reversed of encoder with hidden dimensions $[32, 16, 8]$. We used a channel-wise 2-layer MLP to compute the latent $\mathbf{Z}$ from the encoder's output, and the latent dimension is set to 20. We preprocessed the Wavelet signals of graph data via the PyGSP Defferrard et al. (2017) software. For each graph, we performed Wavelet transform to get its 4-resolution Wavelet tensor, where each scale varies in [0.25, 0.5, 0.75, 1]. Finally, the autoencoder is pretrained in 100 epochs with a batch size of 32 and learning rate of 0.0005.

MoleculeNet 4 shows the hyperparameter settings for fine-tuning MPNN on five downstream datasets in MoleculeNet benchmark. In general, we used local attetion as proposed in Shi et al. (2021). To model the global interactions, we augment virtual nodes to the local models to improve the performances in ToxCast and SIDER.

TUDataset Table 5 summarizes the hyperparameter settings for the transfer learning experiments on six TUDataset benchmark datasets. For IMDB-B and IMDB-M, which lack domain node features, we employed HOPE-WavePE as their initial node features. To create unified node features compatible with the hidden dimensions of the MPNN layers, we added a 2-layer MLP before the MPNN layers to update the combination of domain and HOPE-WavePE features. Following Bodnar et al. (2022), we performed 10-fold validations for each dataset and reported means and standard deviations.

Dataset	# Graphs	# Nodes	# Edges	Pred. level	Pred. task	Metric
CIFAR10	60,000	117.63	469.10	graph	class. (10-way)	ACC
MNIST	70,000	70.57	281.65	graph	class. (10-way)	ACC
ZINC-subset	12,000	23.15	24.92	graph	reg.	MAE
MolBBBP	2,039	24.06	25.95	graph	class. (binary)	AUROC
MolBACE	1,513	34.09	36.86	graph	class. (binary)	AUROC
MolTox21	7,831	18.57	19.29	graph	class. (binary)	AUROC
MolToxCast	8,576	18.78	19.26	graph	class. (binary)	AUROC
MolSIDER	2,039	33.64	35.36	graph	class. (binary)	AUROC
Peptides-func	15,535	150.94	153.65	graph	class. (binary)	AP
Peptides-struct	15,535	150.94	153.65	graph	reg.	MAE
MUTAG	188	17.9	39.6	graph	class. (binary)	ACC
PROTEINS	1,113	39.1	72.8	graph	class. (binary)	ACC
NCI1	4,110	29.9	32.3	graph	class. (binary)	ACC
NCI109	4,127	29.7	32.1	graph	class. (binary)	ACC
IMDB-B	1,000	19.8	96.5	graph	class. (binary)	ACC
IMDB-M	1,500	13.0	65.9	graph	class. (3-way)	ACC

Table 2: Dataset details for transferability experiments on image, ZINC, MoleculeNet, LRGB and TUDataset.

Batch size	# Epoch	Encoder	Decoder	Learning rate	Scales	Latent dim
32	100	[8, 16, 32]	[32,16,8]	0.0005	[0.25, 0.5, 0.75, 1.0]	20

Table 3: Hyperparameter settings for pretraining high-order AE.

ZINC, Image Classification Tasks, and LRGB We follow the best hyperparameter settings issued in previous work of GPS Rampášek et al. (2022) and MPNN+VN Cai et al. (2023); then, we fine-tuned for better performance. Our full hyperparameter studies of the benchmarks are shown in Table 6.

Hyperparameter	**BBBP**	**BACE**	**Tox21**	**ToxCast**	**SIDER**
Pre MPNN	MLP	MLP	MLP	MLP	MLP
MPNN type	Attention	Attention	Attention	Attention	Attention
VN Augmented	-	-	-	✓	✓
# Layers	5	5	5	3	3
Hidden Dim	300	300	300	512	512
Dropout	0.5	0.5	0.5	0.5	0.5
Pooling type	mean	mean	mean	mean	mean
Learning rate	$1e-3$	$1e-3$	$1e-3$	$1e-3$	$1e-3$
Weight decay	$1e-9$	$1e-9$	$1e-9$	$1e-9$	$1e-9$
# Epochs	50	50	50	100	50
Batch size	32	32	32	32	32

Table 4: Hyperparameter settings for downstream evaluations on the MoleculeNet Benchmark.

Hyperparameter	**MUTAG**	**PROTEINS**	**NCI1**	**NCI109**	**IMDB-B**	**IMDB-M**
Node Feat	Domain + PE	Domain + PE	Domain + PE	Domain + PE	PE	PE
Pre MPNN	MLP	MLP	MLP	MLP	MLP	MLP
# MPNN Layers	5	5	5	5	5	5
Hidden Dim	32	32	32	128	128	128
# Epochs	100	100	200	200	100	200
Batch size	128	128	128	128	128	128
Learning rate	1e − 3	1e − 3	1e − 3	1e − 3	1e − 3	1e − 3
Dropout	0.5	0.5	0.5	0.5	0.5	0.5
Graph pooling	mean	mean	mean	mean	mean	mean

Table 5: Hyperameter settings for downstream evaluations on the TUDataset benchmark.

Hyperparameter	**ZINC (subset)**	**MNIST**	**CIFAR10**	**Peptides-func**	**Peptides-struct**
# Layers	9	3	3	3	3
Global Model	Transformer	Transformer	Transformer	Virtual Node	Virtual Node
Local Model	GINE	GatedGCN	GatedGCN	GINE	GINE
Hidden dim	64	50	50	128	128
# Heads	8	4	4	-	-
Dropout	0	0.2	0.2	0	0.01
Graph pooling	sum	mean	mean	mean	mean
PE dim	20	20	20	20	20
Node Update	MLP	MLP	MLP	MLP	MLP
Batch size	128	128	128	32	32
Learning Rate	0.0005	0.003	0.004	0.0005	0.0005
# Epochs	3000	300	300	100	100
# Warmup epochs	50	5	5	5	5
Weight decay	$3e-5$	$2e-4$	$3e-5$	$1e-5$	$1e-5$
# Parameters	452,299	150,081	142,093	477,953	432,206

Table 6: Hyperparameter settings for ZINC, MNIST, CIFAR10, Peptides-func and Peptides-struct dataset