# Report Draft 1

Veljko Kovac, Gerard Planella, Adam Valin,
Luca Pantea, Julius Wagenbach

April 2023

## 1 Introduction

Graph Neural Networks (GNNs), dating back to the 1990s, [Sperduti, 1993, Gori et al., 2005, Bruna et al., 2014, Kipf and Welling, 2016] have ascended from a niche of representation learning to one of its most successful methods. They were designed to analyse graphs commonly used to represent complex relationships and dependencies in various domains, such as social networks, protein structures, and transportation systems. GNNs are powerful tools for analyzing and processing graph data, as they can capture both local and global structural information of a graph. Their applications span a wide range of domains, from recommender systems [Ying et al., 2018] to drug discovery [Han et al., 2021].

A rather significant role in the success of GNNs is partially attributed to the message-passing framework [Gilmer et al., 2017]. In GNNs, each node has an associated feature vector representing its characteristics or attributes. During message passing, nodes send messages to their neighbours based on their own and their neighbours' features. These messages encode information about the node's relevance to the graph. After all messages have been passed, each node updates its feature vector by aggregating the received messages from all neighbours, while maintaining permutation equivariance with respect to the ordering of the nodes. However, it has been shown that the framework suffers from fundamental limitations [Xu et al., 2018], which constrain their expressivity. More specifically, if the aggregator function is chosen to be injective, GNNs are shown to be as powerful as the 1-dimensional Weisfeiler-Lehman (WL) test [Weisfeiler and Leman, 1968, Morris et al., 2018] in distinguishing non-isomorphic (sub-)graphs.

Recent research has been dedicated to enhancing the discriminative capabilities of Graph Neural Network (GNN) models, pushing past the constraints imposed by the Weisfeiler-Lehman (WL) test. In a recent study, the Learnable Structural and Positional Encodings (LSPE) framework was introduced [Dwivedi et al., 2021]. This innovative approach involves incorporating canonical positional information of nodes within GNNs. The adaptable LSPE framework decouples structural and positional representations, enabling them to be learned independently through learnable positional encodings. Drawing inspiration from the Transformer architecture, LSPE integrates Positional Encodings (PE) into the input layer. These encodings are then concatenated with the node's structural features and updated at each subsequent layer using a designated message-passing function. The inclusion of this canonical positioning information bolsters the expressiveness of Message-Passing GNNs.

# 2 Related Works

Positional encodings (PE) are a fundamental concept that significantly influences the effectiveness of many network architectures, including CNNs [Lecun et al., 1998], RNNs [Hopfield, 1982, Hochreiter and Schmidhuber, 1997], and most recently, Transformers [Vaswani et al., 2017], by providing a means to infuse positional information into deep learning models. However, inferring nodes' positions in any given graph is a non-trivial task due to the absence of a canonical positioning method for capturing both local and global information. While GNNs have been shown to outperform traditional algorithms for node classification, they have demonstrated in [Hu et al., 2020] subpar performance when compared to simple heuristics such as Adamic Adar [Adamic and Adar, 2003] on link prediction tasks [Liben-Nowell and Kleinberg, 2003]. This limitation is primarily attributed to GNNs losing track of node identities while conducting intrinsic computations. Recent works [Srinivasan and Ribeiro, 2019, Brüel-Gabrielsson et al., 2022, Wang et al., 2022] have (empirically) rendered the addition of PE in GNNs crucial in achieving state-of-the-art (SOTA) in graph prediction tasks. Several candidates for PE have been proposed, such as Index PE [Murphy et al., 2019], Laplacian Eigenvectors [Dwivedi et al., 2022] and learnable position-aware embeddings based on random anchor node sets [You et al., 2019]. **Todo:** cite RW as well

However, no explicit topological properties have been used in all the aforementioned techniques, where relational structure about the system in a higher dimensional space could be encoded. The Graph Substructure Network (GSN) [Bouritsas et al., 2022] is considered to be the first method that incorporated topology into its architecture, which laid the foundation for many subsequent research papers on topological deep learning [Papillon et al., 2023]. GSN enhanced the message-passing scheme with structural features extracted by (sub-)graph isomorphism.

Furthermore, a novel message-passing procedure was introduced [Bodnar et al., 2021] that focuses on the "lifting" map of induced cycles within a graph. This enables the network to distinguish between rings within atoms on molecular datasets more easily, increasing its expressivity beyond that of the WL test. Finally, E($n$) equivariant Message Passing Simplicial Networks [Eijkelboom, 2023] have combined geometric and topological graph approaches in their architecture. They re-formulated the original EGNN [Kipf and Welling, 2016] to enable learning of features on simplicial complexes. The approach involves lifting the map and including information about the relative positions of communicating simplices in the message-passing formulation, demonstrating the potential for this approach to expand the range of tasks that can benefit from topological deep learning.
**Todo:** paragraph of EGNN

# 3 Methodology

This research project first aims to examine the expressive power of the EGNN model originally proposed by [Satorras et al., 2021] when incorporating both geometrical and topological information throughout the network, compared to the original architecture where only geometrical information (distances between points) is used. As previously mentioned in Section 2, EGNNs leverage the underlying geometry of the space in which the graph lies to improve performance and attain E($n$) equivariance. One particular aspect that renders EGNNs highly effective in capturing graph-level topology yet computationally expensive is that they operate best in fully-connected settings with a moderately deep architecture, which allows them to infer soft adjacency. However, this approach might not always be feasible, as the number of layers needed to learn higher dimensional graph structures (e.g. cliques), scales exponentially, which ultimately renders such architectures intractable for large graphs.

Our initial proposed approach aims to explore the possibility of leveraging a random walk (RW) diffusion-based positional encoding scheme for infusing topological information into the network, in order to learn more representative higher-order topological structures. Random walks [Bourgain, 1985, Chung, 1997] carry considerable descriptive power when considering graph topology, as they effectively capture the graph's diffusion characteristics [Topping et al., 2022]. Formally, the RW matrix can be defined over $k$-steps as:

$$\mathbf{p}_i^{\mathrm{RW}} = \left[\mathrm{RW}_{ii},\ \mathrm{RW}_{ii}^2,\ ...\ \mathrm{RW}_{ii}^k\right] \tag{1}$$

where $\mathbf{p}_i^{\mathrm{RW}} \in \mathbb{R}^k$ is initial PE of node $i$, and $\mathrm{RW} = AD^{-1}$. Inspired by [Dwivedi et al., 2021], we employ a low-complexity approach to the random walk matrix by focusing solely on the probability of node $i$ landing on itself, i.e. $\mathrm{RW}_{ii}$. Similarly to how Transformers [Vaswani et al., 2017] inject positional information, we concatenate PE with the input node features $\mathbf{h}_i^{\mathrm{in}} \in \mathbb{R}^d$, and pass it through a one-layered embedder $\phi_e$ as follows:

$$\mathbf{m}_{ij} = \phi_m\left(\mathbf{h}_i^l,\ \mathbf{h}_j^l,\ \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{2}$$

$$\mathbf{h}_i^{l+1} = \phi_h\left(\mathbf{h}_i^l,\ \sum_{i\neq j}\mathbf{m}_{ij}\right) \tag{3}$$

$$\mathbf{h}_i^0 = \phi_e\left(\begin{bmatrix}\mathbf{h}_i^{\mathrm{in}}\\ \mathbf{p}_i^{\mathrm{RW}}\end{bmatrix}\right) \tag{4}$$

where $\phi_m, \phi_h, \phi_e$ are non-linear functions with learnable parameters. Similarly to [Satorras et al., 2021], we incorporate the relative squared distance between the two node coordinates in the Euclidian space, i.e. $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^3$. However, we fix the particle coordinates throughout training, contrary to what the authors originally proposed.

# 4 Experiments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

## 4.1 Experimental Setup

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

## 4.2 Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

| **Number of layers** | **FC** | **PEinit** | **Test MAE** |
|---|---|---|---|
| 1 layer | × | × | 0.3177 |
| | × | ✓ | 0.2733 |
| | ✓ | × | (Running) |
| | ✓ | ✓ | 0.144 |
| 4 layers | × | × | 0.2682 |
| | × | ✓ | 0.2397 |
| | ✓ | × | (Running) |
| | ✓ | ✓ | 0.1239 |
| 7 layers | × | × | 0.2391 |
| | × | ✓ | 0.2237 |
| | ✓ | × | 0.1044 |
| | ✓ | ✓ | 0.11 |

Table 1: Test MAE by number of layers, FC, and PEinit

## 5 Discussion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

## References

[Adamic and Adar, 2003] Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211–230.

[Bodnar et al., 2021] Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., and Bronstein, M. (2021). Weisfeiler and lehman go cellular: Cw networks. *Advances in Neural Information Processing Systems*, 34:2625–2640.

[Bourgain, 1985] Bourgain, J. (1985). On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52:46–52.

[Bouritsas et al., 2022] Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. (2022). Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668.

[Bruna et al., 2014] Bruna, J., Zaremba, W., Szlam, A., and Lecun, Y. (2014). Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

[Brüel-Gabrielsson et al., 2022] Brüel-Gabrielsson, R., Yurochkin, M., and Solomon, J. (2022). Rewiring with positional encodings for graph neural networks.

[Chung, 1997] Chung, F. R. K. (1997). *Spectral Graph Theory*. American Mathematical Society.

[Dwivedi et al., 2022] Dwivedi, V. P., Joshi, C. K., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. (2022). Benchmarking graph neural networks.

[Dwivedi et al., 2021] Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. (2021). Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*.

[Eijkelboom, 2023] Eijkelboom, F. (2023). E(n) equivariant message passing simplicial networks. In *International Conference on Machine Learning*.

[Gilmer et al., 2017] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212.

[Gori et al., 2005] Gori, M., Monfardini, G., and Scarselli, F. (2005). A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

[Han et al., 2021] Han, K., Lakshminarayanan, B., and Liu, J. Z. (2021). Reliable graph neural networks for drug discovery under distributional shift. *CoRR*, abs/2111.12951.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.

[Hu et al., 2020] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. *CoRR*, abs/2005.00687.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

[Lecun et al., 1998] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Liben-Nowell and Kleinberg, 2003] Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, page 556–559, New York, NY, USA. Association for Computing Machinery.

[Morris et al., 2018] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2018). Weisfeiler and leman go neural: Higher-order graph neural networks. *CoRR*, abs/1810.02244.

[Murphy et al., 2019] Murphy, R. L., Srinivasan, B., Rao, V., and Ribeiro, B. (2019). Relational pooling for graph representations.

[Papillon et al., 2023] Papillon, M., Sanborn, S., Hajij, M., and Miolane, N. (2023). Architectures of topological deep learning: A survey on topological neural networks.

[Satorras et al., 2021] Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR.

[Sperduti, 1993] Sperduti, A. (1993). Encoding labeled graphs by labeling raam. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann.

[Srinivasan and Ribeiro, 2019] Srinivasan, B. and Ribeiro, B. (2019). On the equivalence between node embeddings and structural graph representations. *CoRR*, abs/1910.00452.

[Topping et al., 2022] Topping, J., Giovanni, F. D., Chamberlain, B. P., Dong, X., and Bronstein, M. M. (2022). Understanding over-squashing and bottlenecks on graphs via curvature.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Wang et al., 2022] Wang, H., Yin, H., Zhang, M., and Li, P. (2022). Equivariant and stable positional encoding for more powerful graph neural networks.

[Weisfeiler and Leman, 1968] Weisfeiler, B. and Leman, A. (1968). The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16.

[Xu et al., 2018] Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *CoRR*, abs/1810.00826.

[Ying et al., 2018] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *CoRR*, abs/1806.01973.

[You et al., 2019] You, J., Ying, R., and Leskovec, J. (2019). Position-aware graph neural networks.