The Study of Positional Encodings in GNNs Using Classical Graph Methods

Project code: `github.com/galsal1/Machine-Learning-With-Graphs`

Gal Salman Roey Dahan Noy Netanel Ori Tadela
Tel Aviv University
galsalman@mail.tau.ac.il, roeydahan@mail.tau.ac.il,
noynetanel1@mail.tau.ac.il, oritadela@mail.tau.ac.il

Abstract

Graph Neural Networks (GNNs) have achieved remarkable success in learning from graph-structured data, yet their inherent permutation-invariance often limits their ability to capture node positions beyond local neighborhoods. In this work, we propose incorporating traversal-based positional encoding, specifically Breadth-First Search (BFS) ranks from a maximal degree node, into GNN architectures to enhance their expressiveness. We explore several architectures: raw BFS indices, sinusoidal transformations inspired by transformers, layer-wise BFS injections, and multi-root BFS encodings. We further combine traversal encodings with classical graph-theoretic features such as centrality measures. Through experiments on classification tasks, we show that traversal-based positional encodings improve model performance in graph-level tasks. Moreover, we provide a theoretical proof outline that BFS based GNNs are more expressive than regular GNNs, by introducing a modified version of Weisfeler-Lehman test.

1 Introduction

Graph Neural Networks (GNNs) are a central paradigm for learning over structured data, powering applications in chemistry, biology, recommendation systems, and social networks [6, 8, 10, 15, 16]. By iteratively aggregating information from local neighborhoods, message-passing GNNs have demonstrated strong performance across diverse domains [6].

Despite this success, a fundamental limitation remains: GNNs are inherently *permutation-invariant*, meaning they lack awareness of the global positions of nodes in the graph structure [1, 11]. This restricts their ability to reason about roles, symmetries, and long-range dependencies - key factors in tasks such as graph alignment, and classification, especially on graphs with hierarchical or highly regular structure.

To overcome this, prior work has introduced positional encodings that enrich node features with structural signals beyond local neighborhoods. Spectral methods based on Laplacian eigenvectors [3] capture global information but incur high computational cost ($\mathcal{O}(|V|^3)$) and are unstable under perturbations. Random-walk and diffusion-based approaches [4, 7, 12, 13] provide probabilistic connectivity signals, but often introduce stochastic variance and scalability challenges.

Motivated by these limitations, we propose a deterministic and efficient alternative: traversal-based positional encodings derived from Breadth-First Search (BFS). BFS ranks offer a lightweight, interpretable signal of hierarchical node positions with linear complexity ($\mathcal{O}(|V| + |E|)$), avoiding the instability of stochastic methods and high cost of spectral and stochastic methods. Furthermore, by augmenting BFS with sinusoidal transformations [14] and classical graph-theoretic measures such as eigenvector centrality, we can inject both smooth positional structure and well-understood domain features.

In this work, we systematically study the effectiveness of BFS-based encodings in GNNs. We integrate them into Graph Convolutional Networks (GCNs), and evaluate their impact across diverse graph classification benchmarks, including PROTEINS, NCI1, ENZYMES, IMDB-BINARY, and DD. Our contributions are threefold:

- We introduce traversal-derived encodings (BFS ranks, sinusoidal BFS, multi-root variants, learnable variants, and a variant that combines BFS-GNN with eigenvector centrality) and show how they enrich node representations in GNNs.
- We provide a theoretical characterization of their expressivity via the *WL-BFS* test, demonstrating that BFS-GNNs are strictly more expressive than standard MPNNs (Message Pass-

ing Neural Network).

- We empirically validate our approach, showing consistent improvements in graph-level tasks across five widely used benchmarks.

Overall, our results highlight that traversal-based positional encodings provide a simple, interpretable, and computationally efficient alternative for enhancing structural representation in GNNs.

2 Related Work

The expressiveness of Graph Neural Networks (GNNs) has been shown to improve significantly when nodes are enriched with positional encodings that provide structural context beyond local message passing. Prior work has explored a range of strategies for incorporating such information. Dwivedi et al. [2] introduced Laplacian eigenvector-based encodings that leverage spectral properties of graphs to position nodes in a global coordinate system. Other approaches, such as Position-aware Graph Neural Networks by You et al. [18], employ node-to-landmark distances to capture relative positioning and break the limitations of permutation invariance. While these methods succeed in introducing global structure, they often come at the cost of computational overhead, such as spectral decomposition, or diffusion calculations. Our work differs by introducing a lightweight and deterministic alternative: positional encodings derived from Breadth-First Search (BFS), which maintain the original topology while still injecting meaningful global structure.

Another prominent line of research leverages random walk statistics to capture structural signals. Random Walk Positional Encodings (RWPE) [5] and diffusion-based kernels provide probabilistic measures of node proximity and connectivity that can approximate long-range dependencies. However, since they rely on stochastic traversal, they often introduce variance across runs and can be computationally expensive for large graphs. BFS-based encodings parallel the spirit of these methods by encoding traversal orders and distances, but do so in a deterministic and interpretable way, avoiding the instability of purely random processes while still retaining the ability to represent global connectivity patterns.

Inspiration also comes from natural language processing, where positional encodings are vital for sequence models like transformers. Vaswani et al. [14] introduced sinusoidal embeddings, providing continuous, translation-invariant sequence representations. Recent work extends this to graphs by applying sinusoidal functions to structural features such as distances. Following this line, we apply sinusoidal transformations to BFS-derived ranks, embedding discrete traversal indices into smooth, continuous representations. This approach combines interpretability from traversal with the expressive power of sinusoidal mapping.

Taken together, these works highlight a growing consensus that GNNs benefit from explicit structural bias, whether through spectral methods, stochastic processes, or sequence-inspired embeddings. Our contribution adds to this literature by proposing BFS-based encodings that combine the determinism and interpretability of graph traversal with the expressive richness of sinusoidal embeddings, offering a simple yet effective approach to enhancing structural representation in GNNs.

3 Expressive power of BFS-GNN

In the classical setting of GNNs, the expressivity of a GNN is defined by its ability to generate different embeddings for pairs of graphs. Xu et al. [16] established that every MPNN (Message Passing Neural Network) is bounded by the Weisfeiler-Lehman (WL) test.

We introduce a new test, named *WL-BFS*, show that it is more robust than the WL test, and finally demonstrate that WL-BFS bounds the expressivity of BFS-GNN.

Firstly, let's define the WL-BFS test:

It is immediate that WL-BFS distinguishes between every pair of graphs that WL can distinguish, since the algorithm first applies the standard WL test. Hence, WL-BFS is at least as powerful as WL.

To establish that WL-BFS is strictly more powerful, we provide an example of two graphs that are indistinguishable under the WL test but are distinguishable under WL-BFS.

In figure 1 and 2, we can see G_1, G_2, both are 3-regular graphs, hence, are indistinguishable under regular WL-test. However, we can see that WL-BFS would decree those graphs are different. Hence $WL < WL - BFS$.

Theorem (WL-BFS upper-bounds BFS-GNN). Fix any root–selection rule f mapping each graph G to a vertex $r = f(G)$ of maximal degree (the rule may be deterministic or randomized; it may depend on the WL stable partition). Consider a

Algorithm 1: WL(d) with Single Root in G_1 and Color-Matched Roots in G_2

Input: Graphs G_1, G_2
Output: Indistinguishable or Different
if $\mathrm{WL}(G_1, \deg) \neq \mathrm{WL}(G_2, \deg)$ **then**
  return *Different*;
$\ell_i \leftarrow \mathrm{WL\_colors}(G_i)$ for $i \in \{1, 2\}$;
$\Delta \leftarrow \max \deg(G_1) = \max \deg(G_2)$;
Choose any $r_1 \in V(G_1)$ with $\deg(r_1) = \Delta$;
$c \leftarrow \ell_1(r_1)$;
$R_2 \leftarrow \{\, v \in V(G_2) \mid \ell_2(v) = c \,\}$;
$d_1 \leftarrow \mathrm{BFS\_dist}(G_1, r_1)$;
$S_1 \leftarrow \mathrm{WL}(G_1, d_1)$;
foreach $r_2 \in R_2$ **do**
  $d_2 \leftarrow \mathrm{BFS\_dist}(G_2, r_2)$;
  $S_2 \leftarrow \mathrm{WL}(G_2, d_2)$;
  if $S_1 = S_2$ **then**
    return *Indistinguishable*;
return *Different*;

BFS-GNN that initializes node features by a fixed encoding of $(\deg(v), \mathrm{dist}(r, v))$ and then applies a message-passing GNN (permutation-equivariant updates with a permutation-invariant neighborhood aggregator and a permutation-invariant graph readout). For any graphs G_1, G_2, let $r_i = f(G_i)$ and $d_i(v) = \mathrm{dist}_{G_i}(r_i, v)$. If

$$\mathrm{WL}\big(G_1, d_1\big) \;=\; \mathrm{WL}\big(G_2, d_2\big)$$

(i.e., the stable 1-WL partitions under initialization by d_i and $\deg$ coincide, including class multiplicities), then the BFS-GNN produces identical graph-level outputs on G_1 and G_2. Consequently, WL-BFS is an *explicit upper bound* on the expressivity of BFS-GNN: whenever WL-BFS returns *Indistinguishable* for (G_1, G_2), the BFS-GNN cannot separate them.

Proof. $h_v^{(0)} = \phi\big(\deg(v), d(v)\big)$ for a fixed node-feature encoding (any deterministic ϕ suffices; injectivity on the domain only strengthens the claim). Let the GNN perform T rounds of message passing

$$h_v^{(t+1)} = \Phi^{(t)}\big(h_v^{(t)}, \mathrm{AGG}^{(t)}\{\Psi^{(t)}(h_v^{(t)}, h_u^{(t)}) : u \in N(v)\}\big).$$

where each $\mathrm{AGG}^{(t)}$ is permutation-invariant (e.g.,sum/mean/max) and $\Phi^{(t)}, \Psi^{(t)}$ are shared across nodes; let ρ be a permutation-invariant readout mapping the multiset $\{h_v^{(T)} : v \in V(G)\}$ to a graph embedding.

Run 1-WL in parallel with the same initialization by $(\deg, d)$. Denote by $\chi_v^{(t)}$ the WL color of v after t refinement rounds, and by χ^* the stable partition.

Claim (node-wise simulation). For every $t = 0, \ldots, T$ there exists a function η_t such that

$$\chi^{(t)}_{G_i,d_i}(u) = \chi^{(t)}_{G_j,d_j}(v) \quad \Longrightarrow \quad h_u^{(t)} = h_v^{(t)}.$$

Proof of claim by induction on t. Base $t = 0$: equal WL colors at $t = 0$ mean equal $(\deg, d)$, hence equal $x^{(0)}$, so $h^{(0)}$ match. Step $t \to t + 1$: if $\chi^{(t)}$ are equal for u and v, then (by the WL update rule) the multisets of neighbors' t-round colors coincide. By the induction hypothesis, those colors correspond to identical multisets of neighbor embeddings. Applying the same $\Psi^{(t)}$, the same permutation-invariant $\mathrm{AGG}^{(t)}$, and the same $\Phi^{(t)}$ yields $h_u^{(t+1)} = h_v^{(t+1)}$.

Now assume $\mathrm{WL}(G_1, d_1) = \mathrm{WL}(G_2, d_2)$ at stability. Then there is a bijection between the stable color classes of (G_1, d_1) and (G_2, d_2) preserving class sizes. By the claim with $t = T$, nodes in corresponding classes have identical embeddings; hence the *multisets* $\{h_v^{(T)} : v \in V(G_1)\}$ and $\{h_v^{(T)} : v \in V(G_2)\}$ coincide. Applying the same permutation-invariant readout ρ gives identical graph-level outputs. Therefore any BFS-GNN as specified cannot distinguish G_1 and G_2 when the WL-BFS signatures match.

Corollary (strictness over vanilla GNNs). If there exists a pair (G_1, G_2) such that $\mathrm{WL}(G_1, \deg) = \mathrm{WL}(G_2, \deg)$ but WL-BFS declares *Different*, then some BFS-GNN separates (G_1, G_2) while every vanilla MPNN (bounded by degree-initialized WL) fails. Hence BFS-GNN is strictly more expressive than standard GNNs. BFS on every pair of root nodes s.t. $v_1 \in G_1, v_2 \in G_2$ would generate different distance histogram for each Graph. hence, WL-BFS and BFS-GNN are more expressive than a regular MPNN GNN.

Remark on root selection. The theorem bounds a BFS-GNN with *any* fixed root policy f by the WL run with the corresponding distance initialization $d(v) = \mathrm{dist}(f(G), v)$. The WL-BFS Algorithm chooses r_1 in G_1 and searches all degree/WT-color-matched candidates r_2 in G_2; whenever it finds a match of the WL(d) signatures, the bound

above applies (and shows indistinguishability for any BFS-GNN that could select those roots). If f is randomized, the argument holds conditionally in expectation on sampled roots.

4 Experiments

We evaluate and compare different architectures of BFS-GNN across a variety of datasets and tasks, in order to test the empirical effectivenes of BFS-GNN relative to standard GNNs.

Datasets. We used five graph classification benchmarks: four bioinformatics datasets (PROTEINS, NCI1, DD and ENZYMES) and a social network dataset (IMDB-B). Our objective was to exploit the BFS encoding to capture both global and structural information of graphs, thereby demonstrating its advantage over conventional architectures.

Models and configurations. While experimenting with the integration of BFS encodings, we maintained a consistent underlying network architecture of the form: GCN $\rightarrow$ ReLU $\rightarrow$ GCN $\rightarrow$ Readout $\rightarrow$ Linear $\rightarrow$ Softmax. We used a hidden layer size of 64, and graph-level average readout was performed using global pooling. Each experiment was trained for 50 epochs and repeated 30 times with fixed random seeds to ensure consistent data partitioning and weight initialization across runs. Optimization was carried out using the Adam optimizer [9] with a learning rate of 10^{-3}, and the data was split to 80%, 20% Training and Test accordingly.

For BFS-based architectures, we evaluated the following variants and combinations of them:

- **1-root BFS**: BFS encoding initialized from a maximal degree single root node (tie breaker: minimal node-id).
- **k-root BFS**: concatenated BFS encodings initialized from $k \in \{2, 3, 5, 8, 10\}$ distinct root nodes.
- **sin-BFS**: BFS encodings transformed using trigonometric functions.
- **Learnable-BFS**: BFS encodings passed through a 2-layer perceptron with varying hidden sizes.
- **Centrality-GNN**: Using Spectral Decomposition centrality measures as PE.
- **Layerwise BFS Injection**: BFS encodings concatenated at each GNN layer.

5 Results

We observe consistent gains from BFS encodings over the plain GCN baseline across all five benchmarks (Table 1). The best variant per dataset is: *BFS 2 roots* on PROTEINS (74.6% $\pm$ 1.94), *BFS 5 roots* on NCI1 (71.1% $\pm$ 1.49), *Single-root BFS* on ENZYMES (27.6% $\pm$ 3.67), *Layerwise BFS injection* on IMDB-B (68.6% $\pm$ 20.8), and *Centrality+BFS* on DD (76.5% $\pm$ 6.1). Overall, k-root BFS shows a clear "sweet spot" at $k \in [1, 5]$: modestly increasing k improves robustness (notably on NCI1), while too many roots ($k \geq 8$) dilutes the positional signal and reduce accuracy. The strong IMDB-B result for layerwise injection suggests that repeatedly introducing positional information mitigates over-smoothing on noisier social graphs, while the DD win for spectrality+BFS based PE indicates that BFS loses its edge over big tasks, yet it still contributes to capture the Graph's structure. as the architecture that combined BFS with spectral centrality showed better results than only spectral centrality. Moreover, Training/test curves (Figures 1–3) show faster convergence and lower generalization gap for BFS variants, reduced loss and variance over epochs. Nevertheless, the BFS-GNNs exhibited both steeper learning curves and better generalization results than the vanilla GCNs.

6 Conclusion

We introduced *BFS-GNN*, a positional-encoding scheme based on breadth-first search (BFS) distances. BFS-based encodings are simple, interpretable, and architecture-agnostic, making BFS-GNN a viable addition to any graph-classification network. Theoretically, we proposed a modified Weisfeler-Lehman test, *WL-BFS*, and showed that BFS-GNNs are more expressive than vanilla MPNNs while upper-bounded by WL-BFS. Empirically, BFS variants improve over a plain GCN baseline and centrality encodings on 5 benchmarks. We find a clear sweet spot for multi-root encodings with $k! \in ![2, 5]$, and gains from layerwise reinjection on social graphs. On DD—whose graphs have many more nodes than the other datasets—the mixed variant performs best, suggesting that for bigger graphs, BFS encoding helps capture structure but is dominated by spectral methods. Overall, traversal-derived positional signals provide a cheap and robust way to inject global structure into GNNs, warranting further study on other tasks and different GNN architectures.

Table 1: Dataset statistics and test set accuracies (%). Means and standard deviations are computed over the last 5 epochs as a stability proxy. Best scores per dataset are in **bold**.

Datasets	PROTEINS	NCI1	ENZYMES	IMDB-B	DD
# graphs	1113	4110	600	2000	1000
# classes	2	2	6	2	2
Avg # nodes	39.1	29.8	32.6	19.8	429.6
Models					
Baseline (GCN)	69.3 ± 3.12	66.0 ± 1.54	26.7 ± 3.32	47.8 ± 2.23	70.4 ± 2.34
BFS (single root)	74.5 ± 2.61	70.5 ± 1.08	$\mathbf{27.6 \pm 3.67}$	56.6 ± 3.77	71.1 ± 2.39
Learnable BFS	73.5 ± 3.52	69.7 ± 1.23	26.3 ± 3.23	58.4 ± 3.71	72.3 ± 2.42
Spectral Centrality	71.7 ± 1.48	68.9 ± 1.14	21.9 ± 2.48	57.25 ± 1.66	74.5 ± 1.09
Spectral Centrality + BFS	74.0 ± 2.93	69.8 ± 1.36	24.1 ± 3.38	58.4 ± 3.70	$\mathbf{76.5 \pm 2.49}$
BFS 2 Roots	$\mathbf{74.6 \pm 1.94}$	70.2 ± 1.70	26.3 ± 3.44	61.3 ± 3.28	72.0 ± 2.32
BFS 3 Roots	74.1 ± 2.27	70.7 ± 1.62	26.1 ± 3.02	61.3 ± 2.42	71.7 ± 2.04
BFS 5 Roots	73.5 ± 2.27	$\mathbf{71.1 \pm 1.49}$	25.6 ± 3.18	60.7 ± 2.73	71.1 ± 2.41
BFS 8 Roots	71.7 ± 2.28	71.1 ± 1.56	25.1 ± 3.44	59.5 ± 3.46	70.3 ± 2.20
BFS 10 Roots	71.4 ± 2.22	70.6 ± 1.80	24.3 ± 3.56	60.0 ± 2.92	70.2 ± 2.81
Layerwise BFS Injection	74.3 ± 1.64	69.4 ± 0.97	23.7 ± 2.11	$\mathbf{68.6 \pm 5.00}$	71.0 ± 1.28
Learnable Sinusoidal BFS	73.5 ± 2.79	69.3 ± 1.06	24.0 ± 2.89	60.0 ± 2.92	75.8 ± 2.12

References

[1] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[2] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2110.07875*, 2022.

[3] Vijay Prakash Dwivedi, Chaitanya K. Joshi, et al. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[4] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Ladislav Rampášek, Xavier Bresson, Pietro Liò, Michael Bronstein, and Johannes Klicpera. Long range graph benchmark. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[5] Vijay Prakash Dwivedi, Chaitanya K Joshi, Ladislav Rampášek, Anh Tuan Luu, Yoshua Bengio, and Nils M Kriege. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2022.

[6] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, 2017.

[7] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[8] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[11] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, 2019.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[13] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations (ICLR)*, 2020.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[16] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

[17] Pinar Yanardag and S. V. N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.

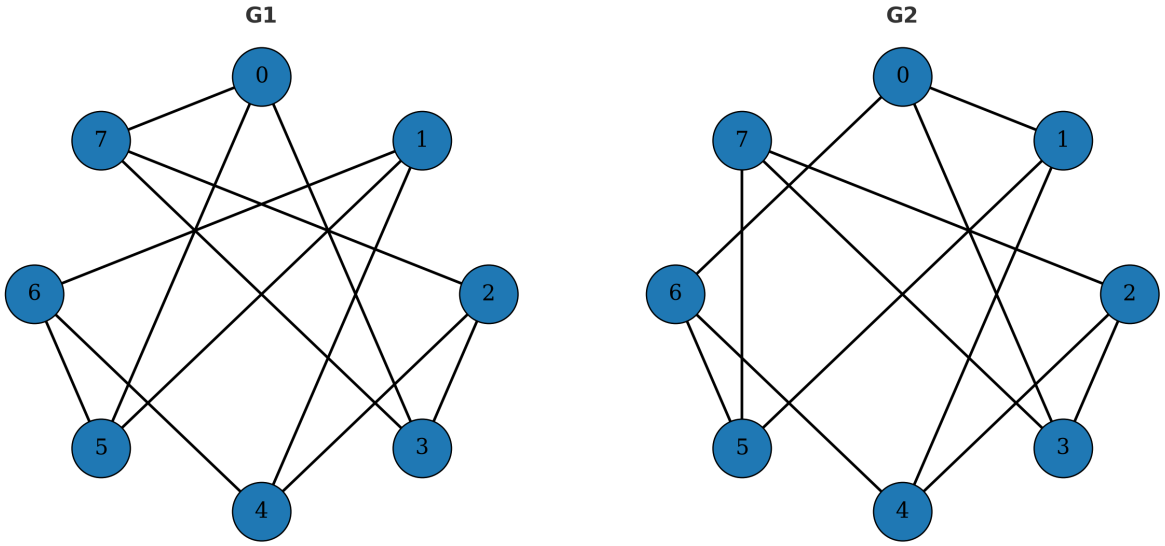
[18] Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 7134–7143. PMLR, 2019.

DETAILS OF DATASETS

We give detailed descriptions of the datasets used in our experiments. Further details can be found in [17].

Social networks dataset. IMDB-BINARY is a movie collaboration dataset. Each graph corresponds to an ego-network for an actor/actress, where nodes represent actors/actresses and an edge is drawn if they appear in the same movie. Each graph is derived from a pre-specified genre of movies, and the task is to classify the genre the graph is derived from.

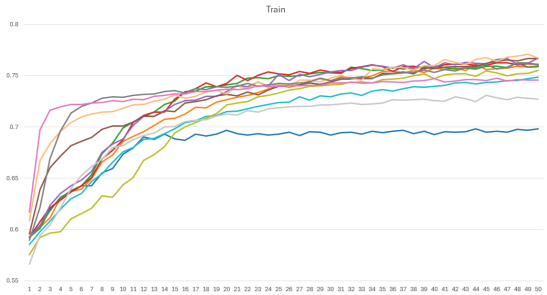
Bioinformatics datasets. PROTEINS is a dataset where nodes are secondary structure elements (SSEs), and edges connect two SSEs if they are neighbors in the amino-acid sequence or in 3D space. The task is to classify proteins into enzymes and non-enzymes. **NCI1** is a dataset made publicly available by the National Cancer Institute (NCI). It consists of chemical compounds screened for their ability to suppress or inhibit the growth of human tumor cell lines. The task is to classify compounds into active or inactive against cancer cell lines. **ENZYMES** is a dataset of protein tertiary structures obtained from the BRENDA database. Each protein is represented as a graph, where nodes correspond to secondary structure elements. The task is to classify proteins into one of the six Enzyme Commission (EC) top-level classes. **DD (D&D)** is a dataset of protein structures, where nodes correspond to amino acids and edges indicate spatial closeness. The task is to classify proteins into enzymes and non-enzymes. This dataset is considerably larger than PROTEINS.



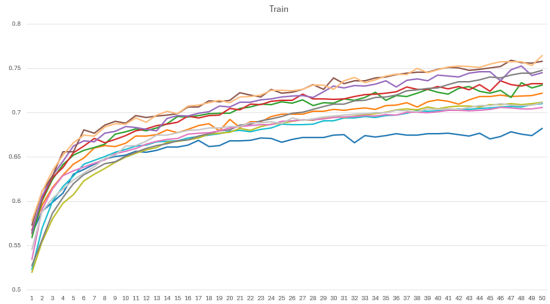
(a) G_1 (3-regular). Indistinguishable under 1-WL, but separated by WL-BFS when rooted at a max-degree node.

(b) G_2 (3-regular). Indistinguishable under 1-WL, but separated by WL-BFS via color-matched rooting.

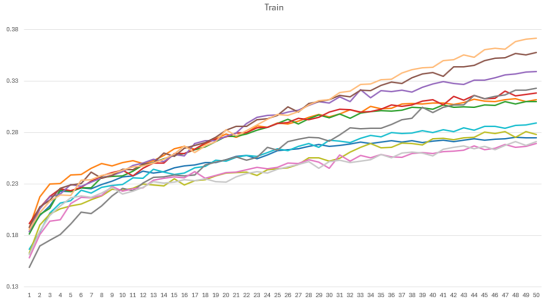
Figure 1: WL-BFS counterexample graphs: **(a)** G_1 and **(b)** G_2.



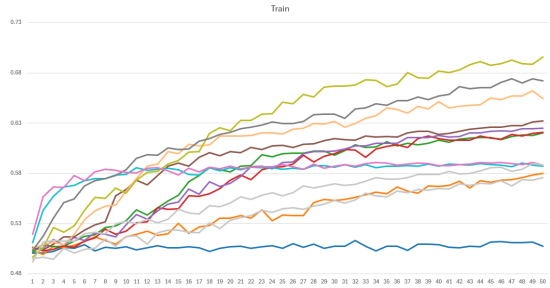
(a) PROTEINS



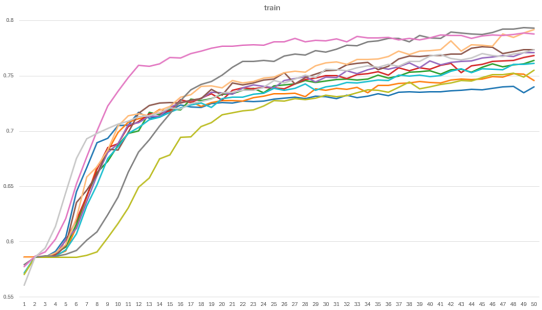
(b) NCI1



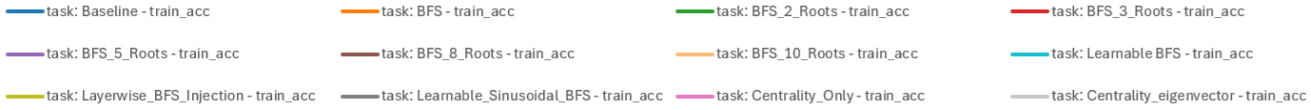
(c) ENZYMES



(d) IMDB-BINARY

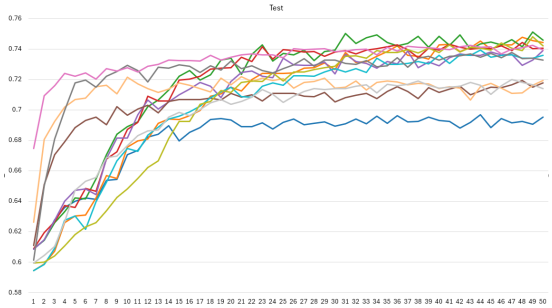


(e) DD

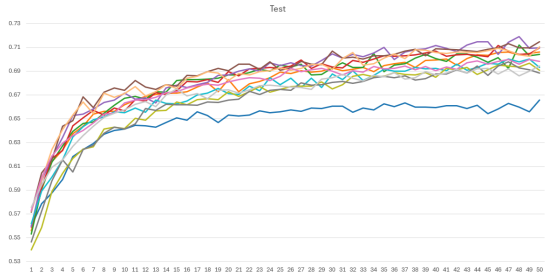


(f) Legend

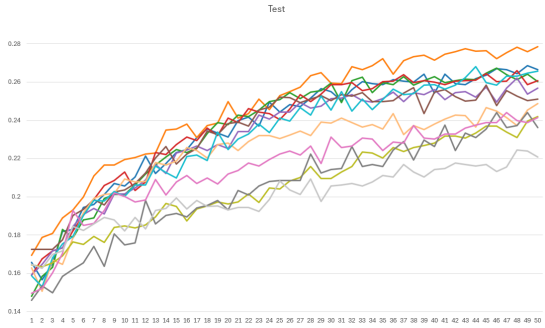
Figure 2: Training set performance of BFS-GNN and baselines across five datasets.



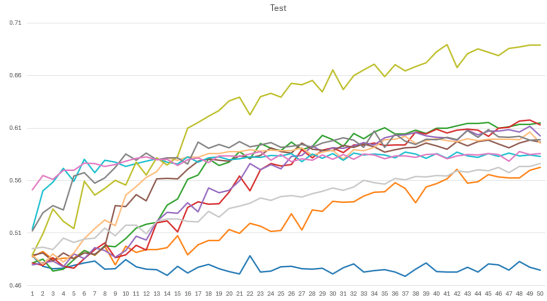
(a) PROTEINS



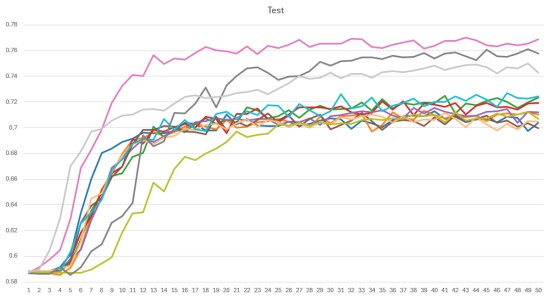
(b) NCI1



(c) ENZYMES



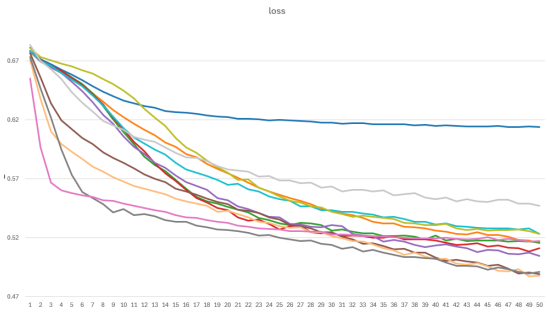
(d) IMDB-BINARY



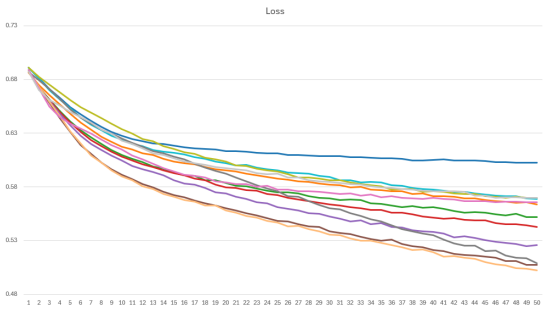
(e) DD

(f) Legend

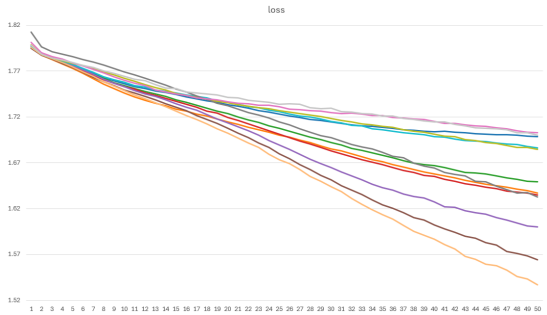
Figure 3: Test set performance of BFS-GNN and baselines across five datasets.



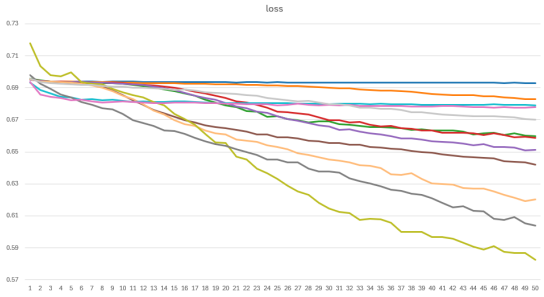
(a) PROTEINS



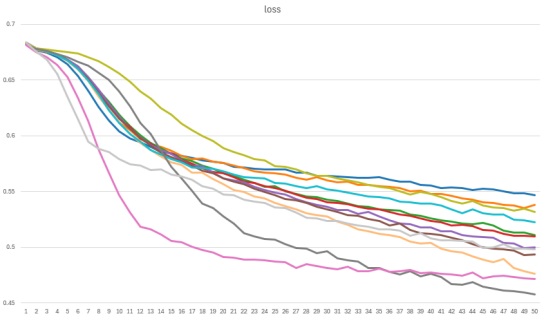
(b) NCI1



(c) ENZYMES



(d) IMDB-BINARY



(e) DD

(f) Legend

Figure 4: loss of BFS-GNN and baselines across five datasets.