
Supplementary Material to Interpretable and Parameter Efficient Graph Neural Additive Models with Random Fourier Features

Anonymous Author(s)

Affiliation

Address

email

1 We organize the content in the supplementary material as follows: In section 1, we present an
2 algorithm for the proposed G-NAMRFF. In section 2, we present proofs that supports permutation
3 equivariance and robustness properties of G-NAMRFF. In section 3, we include an additional discussion
4 on interpretability with node and graph classification tasks. In section 4, we present an ablation
5 studies with respect to the hyperparameters.

6 1 Algorithm for G-NAMRFF

7 In this section, we present an algorithm for G-NAMRFF. Recall, the proposed method models the
8 univariate function on each feature using GP prior. In particular, we proposed the kernel to be both
9 graph and feature aware which is further approximated with RFF features. Leveraging this, we further
10 proposed a light weight model with the number of trainable parameters being number of RFF features.

11 With the input feature matrix as $\mathbf{X} \in \mathbb{R}^{N \times D}$, and normalised graph Laplacian as $\tilde{\mathbf{L}}_{\mathcal{G}}$ the proposed
12 algorithm involves three important steps 1: filtering, 2: computing the RFF mapping and 3: learning
13 the weight vector. Detailed step by step process is presented in Algorithm 1.

Algorithm 1 Algorithm for G-NAMRFF

1: **Inputs:** Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, normalized Laplacian $\tilde{\mathbf{L}}_{\mathcal{G}}$, RFF dimension M , bandwidth Θ , filter order R .
2: **Trainable parameters:** $\{\alpha_h\}_{h=0}^R$, weight vectors $\{\mathbf{w}_k \in \mathbb{R}^M\}_{k=1}^D$.
3: **Output:** Node predictions $\{y_i\}_{i=1}^N$.
4: **Filtering:** Compute filtered features $\tilde{\mathbf{X}} = \sum_{h=0}^R \alpha_h \tilde{\mathbf{L}}_{\mathcal{G}}^h \mathbf{X}$.
5: **RFF computation:** Sample frequencies $\{a_{k,m}\}_{m=1}^M \sim \mathcal{N}(0, 1)$ and phases $\{b_{k,m}\}_{m=1}^M \sim \text{Uniform}(0, 2\pi)$ for each k . Compute feature-wise RFF map: $\Phi_{\mathbf{a}}(x) = \sqrt{\frac{2}{M}} [\cos(a_{k,1}x + b_{k,1}), \dots, \cos(a_{k,M}x + b_{k,M})]^\top$
6: **for** each node $i = 1, \dots, N$ **do**
7: **for** each feature $k = 1, \dots, D$ **do**
8: Obtain feature embedding: $f_k(x_{i,k}) = \Phi_{\mathbf{a}}^\top(\tilde{x}_{i,k}) \mathbf{w}_k$.
9: **end for**
10: **Prediction:** $y_i \leftarrow \sum_{k=1}^D f_k(x_{i,k})$.
11: **end for**
12: **return** $\{y_i\}_{i=1}^N$.

2 Proofs for the Theoretical Characterization of G-NAMRFF

In this section, we present the proofs for the permutation equivariance and robustness properties of G-NAMRFF.

Theorem. 4.1. (Permutation Equivariance) *Let $\mathcal{P} = \{\mathbf{P} \in \{0, 1\}^{N \times N} : \mathbf{P}^\top \mathbf{P} = \mathbf{P} \mathbf{P}^\top = \mathbf{I}_N\}$ be the set of all $N \times N$ permutation matrices. Then under the permutation of the graph Laplacian \mathbf{L}_G and node-feature matrix \mathbf{X} by any $\mathbf{P} \in \mathcal{P}$, the embeddings from G-NAMRFF also modifies as $\mathbf{y}_{\text{perm}} = \mathbf{P} \mathbf{y}$, where $\mathbf{y} \in \mathbb{R}^N$ is the predictions across all the nodes.*

Proof. Let $\mathbf{P} \in \mathcal{P}$ be the permutation matrix acting on the graph \mathcal{G} . Under this permutation, the graph Laplacian and feature matrix transforms as $\tilde{\mathbf{L}}_{\mathcal{G}, \text{perm}} = \mathbf{P} \tilde{\mathbf{L}}_{\mathcal{G}} \mathbf{P}^\top$ and $\mathbf{X}_{\text{perm}} = \mathbf{P} \mathbf{X}$. Let us call the prediction of the node i under the permutation as $y_{i, \text{perm}}$, whereas \mathbf{y}_{perm} is obtained by stacking $y_{i, \text{perm}}$.

Under the permutations, filter output modifies as

$$\begin{aligned} \tilde{\mathbf{X}}_{\text{perm}} &= \sum_{h=0}^R \alpha_h \tilde{\mathbf{L}}_{\mathcal{G}, \text{perm}}^h \mathbf{X}_{\text{perm}} = \sum_{h=0}^R \alpha_h \mathbf{P} \tilde{\mathbf{L}}_{\mathcal{G}}^h \mathbf{P}^\top \mathbf{P} \mathbf{X}, \\ &\stackrel{(a)}{=} \sum_{h=0}^R \alpha_h \mathbf{P} \tilde{\mathbf{L}}_{\mathcal{G}}^h \mathbf{X}, \\ &\stackrel{(b)}{=} \mathbf{P} \tilde{\mathbf{X}}, \end{aligned} \tag{1}$$

where (1)(a), follows from the property of permutation matrix i.e., $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$. Observe that from (1)(b) under the permutation, filtered outputs also gets permuted. Leveraging this, we now evaluate the prediction of G-NAMRFF under permutation as

$$\begin{aligned} \mathbf{y}_{\text{perm}} &= [y_{1, \text{perm}}; \dots; y_{N, \text{perm}}], \\ &= \sum_{k=1}^D \underbrace{[\Phi_{\mathbf{a}}^T(\tilde{x}_{1, k, \text{perm}}); \dots; \Phi_{\mathbf{a}}^T(\tilde{x}_{N, k, \text{perm}})]}_{\Psi_{k, \text{perm}}} \mathbf{w}_k, \\ &\stackrel{(a)}{=} \mathbf{P} \sum_{k=1}^D [\Phi_{\mathbf{a}}^T(\tilde{x}_{1, k}); \dots; \Phi_{\mathbf{a}}^T(\tilde{x}_{N, k})] \mathbf{w}_k, \\ &= \mathbf{P} \mathbf{y}, \end{aligned} \tag{2}$$

where (2)(a), follows from (1)(b) where it can be observed that for a fixed feature k i.e., $\tilde{\mathbf{x}}_{k, \text{perm}} \in \mathbb{R}^{N \times 1} = \mathbf{P} \tilde{\mathbf{x}}_k$, therefore mappings gets reordered with $\Psi_{k, \text{perm}} \in \mathbb{R}^{N \times M} = \mathbf{P} \Psi_k$. Also it can be observed that $\mathbf{z}_{k, \text{perm}} = [f_k(x_{1, k}), f_k(x_{2, k}), \dots, f_k(x_{N, D})] \in \mathbb{R}^{N \times 1} = \mathbf{P} \mathbf{z}_k$ from the permutational equivariance of Ψ_k . \square

Theorem. 4.2. (Robustness to perturbation of graph Laplacian) *Let $\hat{\mathbf{L}}_{\mathcal{G}} = \mathbf{L}_{\mathcal{G}} + \Delta \mathbf{L}_{\mathcal{G}}$ be the Laplacian of the perturbed graph, with $\|\Delta \mathbf{L}_{\mathcal{G}}\|_2 \leq \epsilon$, and assume that the RFF map $\Phi_{\mathbf{a}}(\cdot)$ is C_{RFF} -Lipschitz continuous. Then each node prediction satisfies $|\hat{y}_i - y_i| \leq C K \epsilon D \|X\|_2$, where $C = C_{\text{RFF}} (\max_k \|w_k\|_2)$, $K = \|\alpha\|_1 (R^2 - 1) \left(\frac{R-1}{R+1}\right)^R$ are constants.*

Proof. Before, proceeding to the proof, we state the following Lemma that bounds the error between the FIR filters built with $\mathbf{L}_{\mathcal{G}}$ and $\hat{\mathbf{L}}_{\mathcal{G}}$. With slight abuse of notation from here on we represent normalized Laplacian with $\mathbf{L}_{\mathcal{G}}$ instead of $\tilde{\mathbf{L}}_{\mathcal{G}}$.

Lemma 1. [3] *Consider a polynomial filter $\mathbf{H}(\mathbf{L}_{\mathcal{G}}) = \sum_{h=0}^R \alpha_h \mathbf{L}_{\mathcal{G}}^h$, and perturbed Laplacian as $\hat{\mathbf{L}}_{\mathcal{G}} = \mathbf{L}_{\mathcal{G}} + \Delta \mathbf{L}_{\mathcal{G}}$ with $\|\Delta \mathbf{L}_{\mathcal{G}}\|_2 \leq \epsilon$ then*

$$\|\mathbf{H}(\mathbf{L}_{\mathcal{G}}) - \mathbf{H}(\hat{\mathbf{L}}_{\mathcal{G}})\|_2 \leq \frac{1}{4} \|\alpha\|_1 (R^2 - 1) \left(\frac{R-1}{R+1}\right)^R \epsilon, \tag{3}$$

where $\|\alpha\|_1 = [\alpha_0, \dots, \alpha_R] \in \mathbb{R}^{R+1}$.

Lemma 1 shows that the error between the outputs of two filters fed with true and perturbed Laplacian is bounded and scales linearly with the error norm. We leverage the above Lemma to bound the difference between the prediction obtained from G-NAMRFF under perturbations.

Assume that the prediction of node i under the graph perturbation as \hat{y}_i . To begin with we evaluate the difference between the filter outputs. Considering the filter outputs with and without perturbation of graph as $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}_{\text{per}}$. We have

$$\begin{aligned}\tilde{\mathbf{X}} &= \mathbf{H}(\mathbf{L}_{\mathcal{G}})\mathbf{X}, \\ \tilde{\mathbf{X}}_{\text{per}} &= \mathbf{H}(\hat{\mathbf{L}}_{\mathcal{G}})\mathbf{X}.\end{aligned}\tag{4}$$

Then the difference between the filtered outputs are bounded as

$$\begin{aligned}\|\tilde{\mathbf{X}}_{\text{per}} - \tilde{\mathbf{X}}\|_2 &= \|\mathbf{H}(\hat{\mathbf{L}}_{\mathcal{G}})\mathbf{X} - \mathbf{H}(\mathbf{L}_{\mathcal{G}})\mathbf{X}\|_2 \\ &\stackrel{(a)}{\leq} \|\mathbf{H}(\hat{\mathbf{L}}_{\mathcal{G}}) - \mathbf{H}(\mathbf{L}_{\mathcal{G}})\|_2 \|\mathbf{X}\|_2 \\ &\stackrel{(b)}{\leq} \frac{1}{4} \|\alpha\|_1 (R^2 - 1) \left(\frac{R-1}{R+1}\right)^R \epsilon \|\mathbf{X}\|_2,\end{aligned}\tag{5}$$

where (5)(a) from a norm inequality and (5)(b) follows from (3). From (5)(b) it is clear that the difference of filter outputs is bounded linearly with the energy in the error term. Leveraging this we now evaluate the difference in the prediction output. We index the i, k element of $\tilde{\mathbf{X}}_{\text{per}}$ with $\tilde{x}_{i,k,\text{per}}$. Then,

$$\begin{aligned}|\hat{y}_i - y_i| &= \left| \sum_{k=1}^D \Phi_a^T(\tilde{x}_{i,k,\text{per}}) \mathbf{w}_k - \sum_{k=1}^D \Phi_a^T(\tilde{x}_{i,k}) \mathbf{w}_k \right| \\ &\leq \sum_{k=1}^D \|\Phi_a^T(\tilde{x}_{i,k,\text{per}}) - \Phi_a^T(\tilde{x}_{i,k})\|_2 \|\mathbf{w}_k\|_2 \\ &\stackrel{(a)}{\leq} C_{\text{RFF}} \sum_{k=1}^D |(\tilde{x}_{i,k,\text{per}}) - (\tilde{x}_{i,k})| \|\mathbf{w}_k\|_2 \\ &\stackrel{(b)}{\leq} \sum_{k=1}^D C_{\text{RFF}} \max_k \{\|\mathbf{w}_k\|_2\} \underbrace{\frac{1}{4} \|\alpha\|_1 (R^2 - 1) \left(\frac{R-1}{R+1}\right)^R}_{K} \epsilon \|\mathbf{X}\|_2 \\ &\stackrel{(c)}{\leq} CK \epsilon D \|\mathbf{X}\|_2,\end{aligned}\tag{6}$$

where (6)(a) follows from Lipschitz continuity of Φ_a , (6)(b), follows from (5)(b). Whereas, (6)(c) follows by considering the $C = C_{\text{RFF}} \max_k \{\|\mathbf{w}_k\|_2\}$. From (6)(c), it is clear that difference in the node predictions scales linearly with the norm of the error term $\Delta \mathbf{L}_{\mathcal{G}}$.

□

3 Additional Discussion on Interpretability

3.1 Interpretability on Node Classification Task

In this section, we present additional examples of the learned univariate functions on the PubMed dataset for the node classification task. Specifically, in Fig. 1, we illustrate the univariate function outputs for the features i.e., keywords as *insulin*, *fat*, *liver*, and *diet*. Recall that at any given feature value, the curve with the highest output indicates the class that the feature most strongly supports.

In Fig. 1a, we examine the influence of the presence of *diet* keyword in the document on the model's predictions. It can be observed that the model associates higher output with type 2 diabetes, suggesting that *diet* is a strong predictor for this class. This observation aligns well with existing medical literature, which indicates that type 2 diabetes is heavily influenced by dietary factors, whereas type 1 diabetes, being an autoimmune condition is less affected by diet [5]. Consistent with

69 this, the model assigns less importance to type 1 diabetes for this feature. In Fig. 1b, we plot the
 70 influence of *fat* keyword in the document where it is clear that it contributes less significantly to the
 71 prediction of type 1 diabetes, further supporting the model’s ability to capture medically relevant
 72 patterns. Similarly, in Fig. 1c and Fig. 1d we plot the impact of these features on model prediction
 73 where it is clear that presence of this word aligns prediction towards type 2 and gestational diabetes.
 74 Although for illustration purposes we have presented few examples one can follow a similar procedure
 75 to obtain the contribution of different features on different datasets.

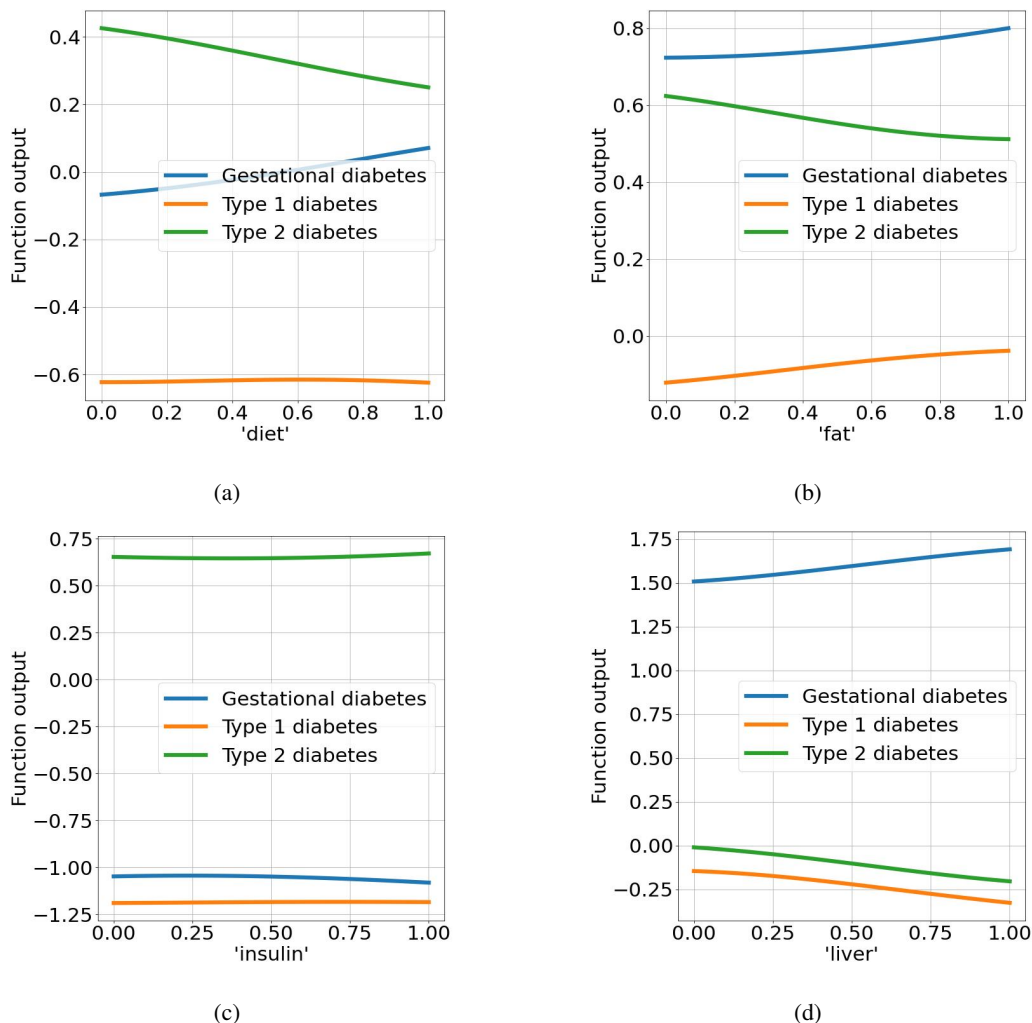


Figure 1: PubMed Dataset: Univariate function outputs on features

76 3.2 Interpretability on Graph Classification Task

77 In this section, we include more discussion on interpretability on the mutagenicity dataset. Here
 78 the main target is to give local level (structural) explanations and also compare them with the
 79 GNN-Explainer [7] which is a *post-hoc* explainer.

80 With the focus on mutagenic class, we give structure level explanations for the model prediction. In
 81 Figs. 2a, 2c and 2e we present three example molecules that are predicted as mutagenic and highlight
 82 predicted substructures responsible for the mutagenic prediction. It is clear that the NH_2 , NO_2 and
 83 aromatic rings act as key contributors to mutagenic class. These explanations align with the existing
 84 studies which advocate the presence of these substructures highly influences the mutagenicity [1].
 85 For a fair comparison we present the substructures revealed from GNN-Explainer in Fig. 2b, 2d

86 and 2f. It can be observed that compared to GNN-Explainer, our proposed model identifies NH_2 ,
 87 NO_2 and aromatic rings more precisely.

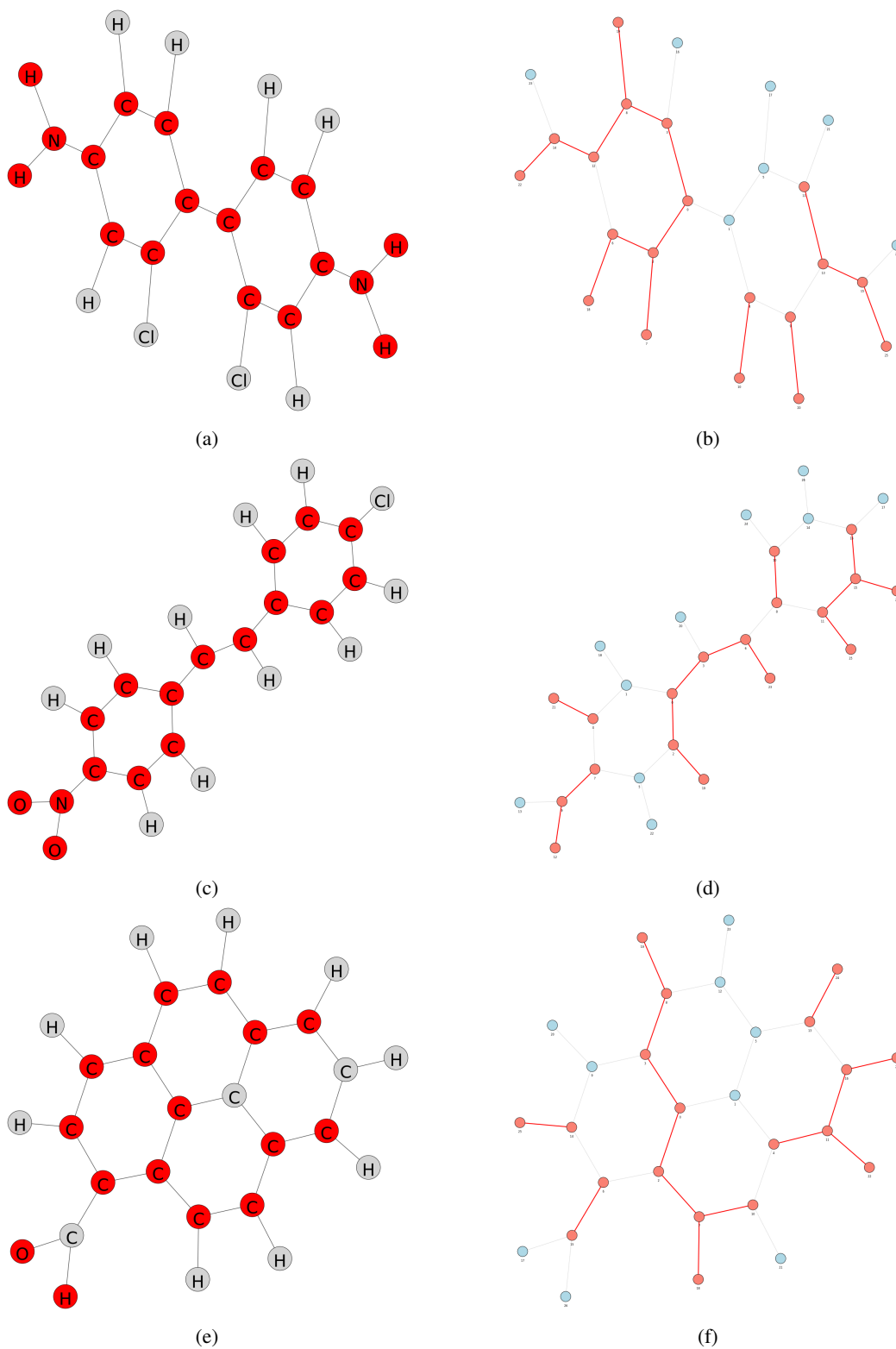


Figure 2: Mutagenicity Dataset: Local level explanations

Dataset	Number of RFFs (M)	Filter Order (R)	Kernel width (Θ)	Learning rate (Lr)	Weight Decay (Wd)
Cora	100	7	2.2	1.8e-2	2.4e-4
Citeseer	100	5	2.5	5.1e-2	3.1e-4
Pubmed	100	5	3.8	2.7e-3	4e-4
Cornell	100	6	2.9	7.3e-2	3.7e-4
ogbn-arxiv	100	4	3.4	5.2e-2	1.3e-4
ogbn-products	40	5	2.3	2.9e-2	5e-4
Proteins	50	5	1.9	3.2e-3	5e-3
Mutag	200	7	1.2	3.7e-2	5e-3
Mutagenicity	200	6	1.4	1.1e-2	5e-4
NCII	50	7	3.7	1.6e-2	5e-4
PTC	50	5	1.7	2.9e-3	2.5e-5

Table 1: Hyperparameter details

4 Hyperparameter Details and Ablation Studies

In this section, we first present the hyperparameter details used in our node and graph classification tasks. Then we analyze how each of these hyperparameters affects G-NAMRFF performance across datasets.

4.1 Hyperparameter Details

We have conducted all the experiments using an *NVIDIA* A30 GPU. In particular, we set the tuning range as follows: Number of RFFs (M)- $\{20, 40, 50, 100, 200\}$, filter order (R) - $[1, 7]$, kernel width (Θ)- $[1.0, 4.0]$, learning rate (Lr)- $[1e-4, 2e-1]$ and weight decay (Wd)- $[1e-5, 1e-2]$. The hyperparameter configurations that yielded the best validation performance are detailed in Table 1. For node classification tasks, we follow the standard dataset splits as specified in [2, 4, 6]. In the case of graph classification tasks, where no standard splits are available, we employ 10-fold cross-validation. All models are trained for 1000 epochs using the *AdamW* optimizer with randomly initialized parameters. The specific learning rates and weight decay values used across different experiments are also reported in Table 1. For node classification, we report the mean classification accuracy computed over five independent random seeds. For graph classification, we report the mean classification accuracy across 10 folds, with each fold averaged over three random seeds.

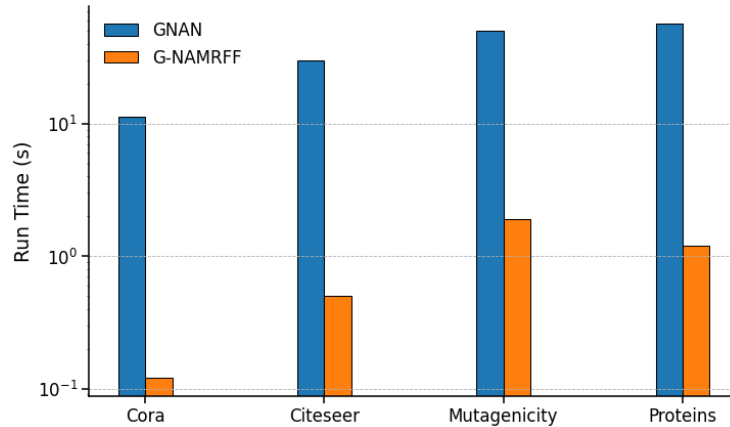
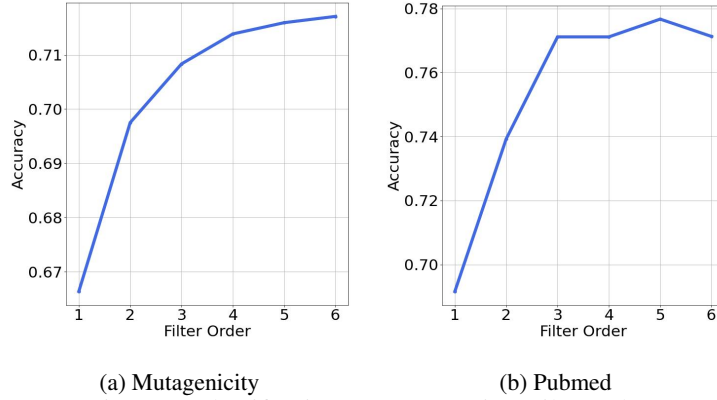


Figure 3: Run time comparison

4.2 Run time analysis

In this section, we compare the runtime performance of G-NAMRFF, against GNAN, which is the only existing *glass-box* GNN architecture. In Fig. 3, we present the per-epoch runtime comparison on both node and graph classification tasks. Notably, it can be observed that even when GNAN is configured



(a) Mutagenicity (b) Pubmed
Figure 4: Classification accuracy against Filter order

with a relatively small hidden dimension of 32, it still requires 10 to 100 times more runtime per epoch compared to G-NAMRFF.

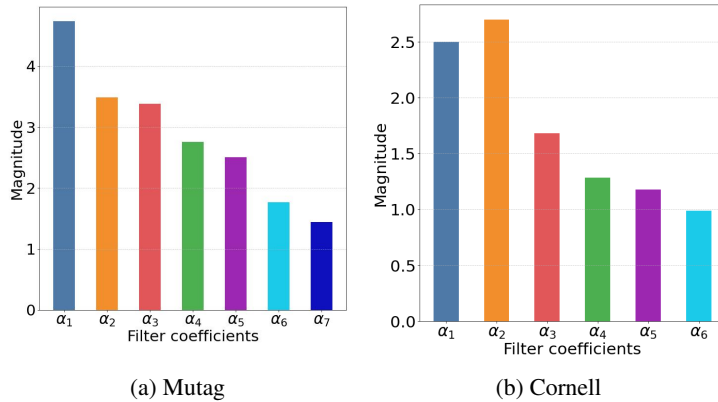
Furthermore, it is worth emphasizing that on small-scale datasets such as Citeseer and Pubmed, increasing the hidden dimension to 64 can already result in out-of-memory (OOM) errors when training GNAN. On large-scale datasets like ogbn-products, training GNAN becomes entirely infeasible due to excessive memory requirements.

4.3 Ablation studies

In this section, we analyze the impact of hyperparameters on the model performance. Recall the key hyperparameters in G-NAMRFF includes the filter order (R) and number of RFF (M).

Filter Order: In Fig. 4, we illustrate the effect of the filter order on classification accuracy for two tasks: node classification on the Pubmed dataset and graph classification on the Mutagenicity dataset. Recall that a filter of order R aggregates information from R -hop neighborhoods in the graph. It can be observed that as the filter order increases, the model is able to incorporate information from larger receptive fields, leading to an initial improvement in classification accuracy. However, beyond a certain point, further increasing the filter order causes the accuracy to saturate or decline, likely due to oversmoothing. Notably, we observe that using higher-order filters ($R > 1$) significantly improves performance compared to $R = 1$, highlighting the benefit of aggregating information from larger graph neighborhoods.

To further highlight the relative importance of different node neighborhoods, we present the magnitudes of the learned filter coefficients for a fixed filter order in Fig. 5, using the Mutagenicity and Cornell datasets. It can be observed that the model assigns higher weights to closer neighborhoods, emphasizing the greater importance of immediate node interactions in the classification task.



(a) Mutag (b) Cornell
Figure 5: Filter coefficients magnitude against filter order

Random Fourier Features:

We analyze the effect of the number of RFFs on classification accuracy. Figs. 6a, 6b, and 6c show the performance variation on the Cornell, PubMed, and NCI1 datasets, respectively. As the number of random Fourier features (M) increases, the classification accuracy initially improves due to better kernel approximation. However, beyond a certain point, performance drops as the model may begin to overfit with larger M . Across all datasets, we observe that good performance can be achieved with relatively small values of M , which translates to a lower number of learnable parameters. Recall that the total number of parameters in G-NAMRFF is $D \times M + R + 1$. As observed M being small the model remains lightweight while being effective.

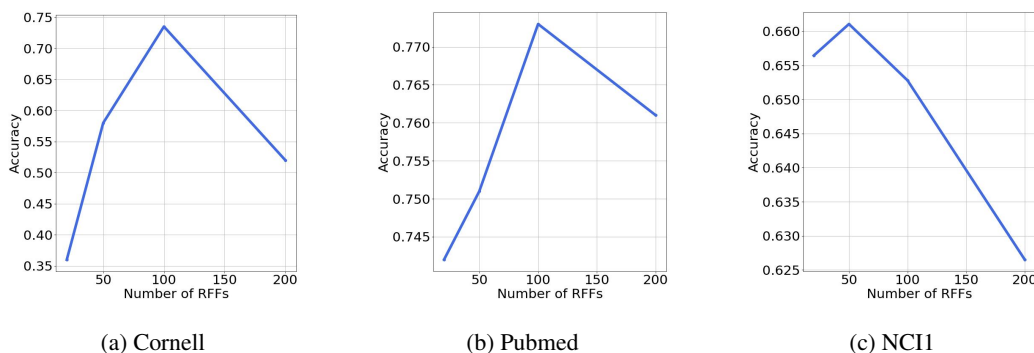


Figure 6: Classification accuracy vs Number of RFFs

References

- [1] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.
- [2] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.
- [3] Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of polynomial spectral graph filters. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5350–5354. IEEE, 2020.
- [4] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [5] Anna-Maria Lampousi, Sofia Carlsson, and Josefin E Löfvenborg. Dietary factors and risk of islet autoimmunity and type 1 diabetes: a systematic review and meta-analysis. *EBioMedicine*, 72, 2021.
- [6] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.
- [7] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.