GRAPH KERNEL CONVOLUTIONS FOR INTERPRETABLE CLASSIFICATION

Magdalena Proszewska
University of Edinburgh, UK
m.proszewska@ed.ac.uk

N. Siddharth
University of Edinburgh, UK
The Alan Turing Institute, UK
n.siddharth@ed.ac.uk

ABSTRACT

State-of-the-art Graph Neural Networks (GNNs) have demonstrated remarkable performance across diverse domains, hence the growing demand for more interpretable GNN techniques. While current research predominantly centers on post hoc perturbation techniques, recent studies propose use of Graph Kernel Convolutions (GKConv) to increase GNNs interpretability intrinsically. These models employ trainable graph filters for extracting hidden features, yet their interpretability is limited since they heavily rely on multilayer perceptrons (MLPs) to make the final predictions. We argue that the latter is not necessary and it is possible to build a model that solely relies on graph kernels and a simple linear layer. Additionally, we integrate contrastive loss to encourage the learning of a more descriptive set of graph filters. In consequence, its decision-making process described through found graph filters and said linear layer is more interpretable. As a proof of concept, we propose a shallow GKConv Interpretable Classifier, which is able to achieve state-of-the-art results while exhibiting better interpretability.

1 INTRODUCTION

Graph Neural Networks (GNNs) (Kipf & Welling, 2017) serve as a versatile model for graph-based applications, employing Message Passing Neural Networks (MPNNs) as its foundation (Gilmer et al., 2017). Current approaches to their interpretability often use post hoc perturbation methods to extract relevant subgraphs (Ying et al., 2019). While insightful post-training, they may fail to capture true decision processes, posing misinterpretation risks. Recent studies (Nikolentzos & Vazirgiannis, 2020; Cosmo et al., 2021; Feng et al., 2022) propose Graph Kernel Convolutions (GKConv) to enhance GNNs interpretability. They employ trainable graph filters to capture patterns in the data, yet their interpretability is limited since they heavily rely on multilayer perceptrons (MLPs) to make predictions based on kernel responses and optionally input features. MLP's complexity, as opposed to a single-layer network, makes it difficult to capture basic linear relationships between inputs and outputs, and its use in GKConv model worsens its transparency. Our work seeks to overcome this limitation and improve GKConv interpretability, particularly in classification tasks. We argue that Graph Kernels are a powerful tool with the potential to be a breakthrough in GNNs interpretability. We propose Graph Kernel Convolution Interpretable Classifier (GKConvIC) that demonstrates the interpretability capabilities of GKConv, simultaneously achieving state-of-the-art accuracy.

2 METHODOLOGY

Let $G = (V, E)$ be an input graph and let G_v represent k-hop neighborhood of $v \in V$ for $k \in \mathbb{N}_+$. Let $K : \mathbb{G} \times \mathbb{G} \to \mathbb{R}$ be a graph kernel that operates on pairs of graphs from the set $\mathbb{G}$ and yields real-valued scores representing the similarity between them.

Graph Kernel Convolution For a graph kernel K and a set of graph filters $\mathcal{F} = \{F_i\}_{i \in I}$, Graph Kernel Convolution is defined as

$$\mathrm{GKConv}(G; \mathcal{F}) = \left[K(G_v, F_i)\right]_{v \in V, i \in I} \in \mathbb{R}^{|V| \times |I|}. \tag{1}$$

The objective in GKConv training is to find a set of optimal graph filters in regard to a given loss function, specifically their adjacency matrices and node feature matrices. For a kernel K differentiable in respect to F_i, it can be done using gradient descent (Feng et al., 2022). We denote a directly differentiable GKConv as DiffGKConv. For non-differentiable kernels and discrete representation of

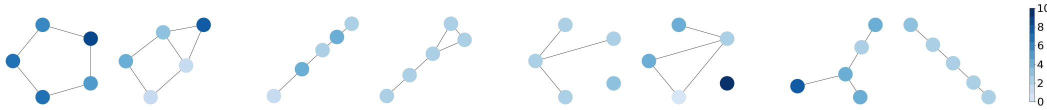


Figure 1: Pairs of graph filters discovered by DiscGKConvIC+GS, DiffGKConvIC+RW, DiffGKConvIC+RW with MLP instead of $\mathbf{W}$, DiffGKConvIC+RW without $\mathcal{L}_{CTR}$, respectively, trained to distinguish graphs with a five-node cycle or a house motif.

graphs (DiscGKConv), Cosmo et al. (2021) uses Discrete Randomized Descent (DRD) strategy for backpropagation. During backpropagation step, an edit operation of each graph filter (add/remove edge, change node label) is sampled and accepted only if loss did not increase. The probability distribution over the edit operations is also optimized using the same gradient estimation.

GKConv Interpretable Classifier We define a classifier GKConvIC that during the forward step: 1) Encodes an input graph G into a hidden representation $\mathbf{Z} \in \mathbb{R}^{|V| \times |I|}$ using GKConv with trainable graph filters $\mathcal{F}$, BatchNorm and ReLU, consecutively. 2) Aggregates node embeddings into a graph embedding denoted as $\hat{z} = \text{agg}(\mathbf{Z}) \in \mathbb{R}^{r \cdot |I|}$, where $r \in \mathbb{N}_+$ is the number of aggregation functions applied along node dimension. 3) Outputs class prediction logits $\hat{z}^T \mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{r \cdot |I| \times C}$ represents weights of the last layer for C classes. Diagram in Appendix A.1 illustrates these steps.

We force the model to find filters $\mathcal{F}$ that allow it to distinguish between classes and make interpretable prediction by incorporating contrastive loss and a specialized initialization of $\mathbf{W}$. Let G be of class y. Let I_y denote indexes of filters of class y i.e. $1/C$ of I. Let $\mathbf{Z}_y = [\mathbf{Z}_{vi}]_{v \in V, i \in I_y}$, $\mathbf{Z}_{\neg y} = [\mathbf{Z}_{vi}]_{v \in V, i \notin I_y}$ denote kernel responses from filters of class y and other classes, respectively. Let $\sigma(x) = \exp(x/\tau)$ for $\tau \in \mathbb{R}_+$, $\mathbf{Z}'_y = \sigma(\mathbf{Z}_y)$ and $\mathbf{Z}'_{\neg y} = \sigma(\mathbf{Z}_{\neg y})$. We define a contrastive loss

$$\mathcal{L}_{CTR}(\mathbf{Z}, y) = -\log \frac{\max \mathbf{Z}'_y}{\sum \mathbf{Z}'_{\neg y} + \max \mathbf{Z}'_y}, \tag{2}$$

where $\sum$ and $\max$ are applied across all elements of the matrix. It encourages model to find at least one filter of class y that gives a strong response to one of the subgraphs G_v, while pushing filters of other classes away. Moreover, we initialize $\mathbf{W}$ so that connections between aggregated kernel responses for filters of a given class and its corresponding logit are set to a positive value, while cross-class connections are set to a negative value. The training loss is as follows:

$$\mathcal{L} = \mathcal{L}_{CE}(\text{agg}(\mathbf{Z})^T \mathbf{W}, y) + \lambda \mathcal{L}_{CTR}(\mathbf{Z}, y), \tag{3}$$

where $\mathcal{L}_{CE}$ denotes the cross entropy loss and $\lambda \in \mathbb{R}_+$ balances two loss components[1].

3 Experiments

Sanity check experiment on a synthetic dataset For the BA-2motifs dataset (Luo et al., 2020), GKConvIC models are expected to find graph filters corresponding to a five-node cycle and a house motif, which define classes. Furthermore, using $\mathbf{W}$ instead of MLP, and $\mathcal{L}_{CTR}$ should yield more descriptive filters. We train 4 models: DiscGKConvIC with Graphlet Kernel (GS), DiffGKConvIC with RW Kernel, and versions of the latter with MLP instead of $\mathbf{W}$, and without $\mathcal{L}_{CTR}$. Each with 2 graph filters of size 5 and agg $=$ mean. All achieved accuracy of 98%-100%. Filters visualized in Figure 1 confirm that both our models are indeed able to discover relevant filters, while the ones found by alternate versions are less suited. Moreover, they learned $\mathbf{W} \in \mathbb{R}^{2\times 2}$ such that $\mathbf{W}_{ii} > 0$ for $i = 1, 2$ and $W_{ij} < 0$ for $i \neq j$, thus affirming GKConvIC effectiveness and interpretability.

Ablation study We study the influence of aggregation functions, the contrastive loss and the initialization of the last layer $\mathbf{W}$. The results in Appendix B.1 show the significance of aggregation functions and the advantages of the contrastive loss. While our initialization of $\mathbf{W}$ has a minor impact on final accuracy, visualizations of learned weights illustrate its influence on interpretability.

Classification accuracy We compare GKConvIC performance against other GKConv models. The results in Appendix B.2 show that we achieve accuracy on the state-of-the-art level, while not relying on MLPs, hence providing more interpretability. DiscGKConvIC performance is slightly worse, which we attribute to its unstable backpropagation technique.

4 Conclusions

In this paper, we proposed GKConv Interpretable Classifier to demonstrate potential of graph kernels and Graph Kernel Convolutions for GNNs interpretability. Our experiments show that GKConvIC is able to achieve state-of-the-art accuracy while exhibiting high level of interpretability.

[1] Code available at `https://github.com/mproszewska/gkconvic`.

References

K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21 Suppl 1:i47–56, 01 2005.

Luca Cosmo, Giorgia Minello, Michael Bronstein, Emanuele Rodolà, Luca Rossi, and Andrea Torsello. Graph kernel neural networks, 2021.

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. doi: 10.1021/jm00106a046. URL https://doi.org/10.1021/jm00106a046.

Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification, 2022.

Aosong Feng, Chenyu You, Shiqiang Wang, and Leandros Tassiulas. Kergnns: Interpretable graph neural networks with graph kernels, 2022.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network, 2020.

Giannis Nikolentzos and Michalis Vazirgiannis. Random walk graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16211–16222. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ba95d78a7c942571185308775a97a3a0-Paper.pdf.

Ida Schomburg, Antje Jäde, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: Updates and major new developments. *Nucleic acids research*, 32:D431–3, 01 2004. doi: 10.1093/nar/gkh081.

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*, KDD '15, pp. 1365–1374, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783417. URL https://doi.org/10.1145/2783258.2783417.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks, 2019.

A METHOD

A.1 GKCONV INTERPRETABLE CLASSIFIER

Diagram in Figure 2 illustrates our method.

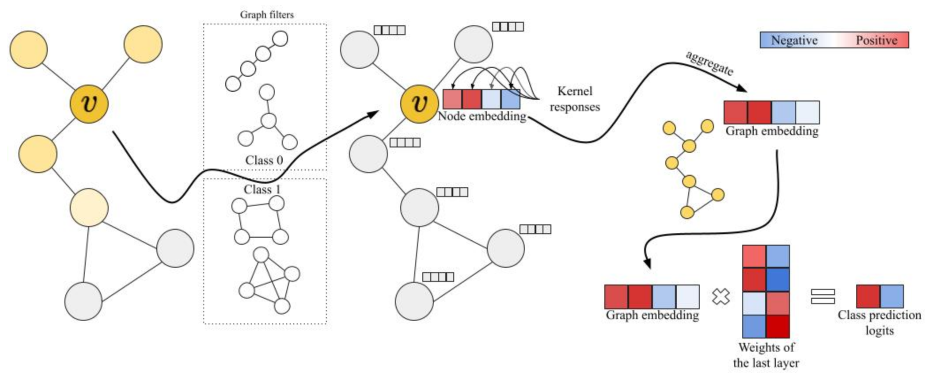
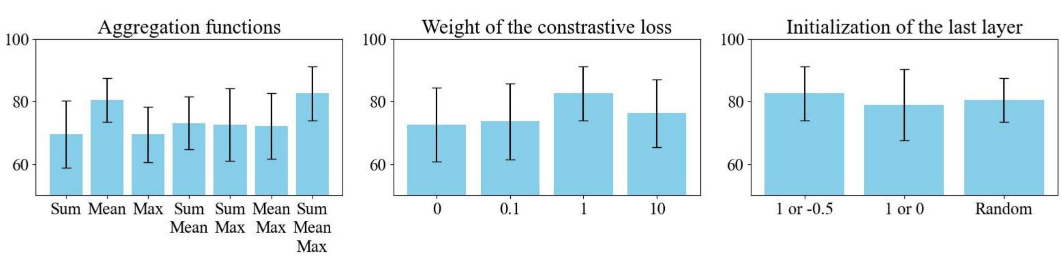


Figure 2: Diagram of GKConv Interpretable Classifier: 1) Node features update to kernel responses between the 2-hop neighborhood and graph filters. 2) Aggregated node embeddings form a graph embedding. 3) Graph embedding is passed through the last layer $\mathbf{W}$, providing the class prediction.

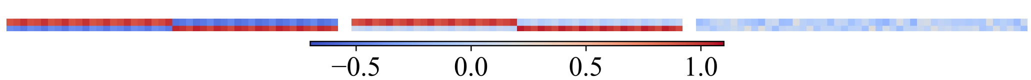
B EXPERIMENTAL RESULTS

B.1 ABLATION STUDY

For our ablation study, we consider MUTAG dataset (Debnath et al., 1991). We use DiffGKConv version of the model since its training is much faster and more stable than DiscGKConv, and set number of graphs filters to 16 and their size to 6. We set input graph subgraphs to be 2-hop neighborhoods with maximum size of 10. We study influence of aggregation functions, the contrastive loss and the initialization of the last layer $\mathbf{W}$. In the baseline configuration, we use 3 aggregation functions (sum, mean, max), contrastive loss weight λ equal 1, and initialize the last layer $\mathbf{W}$ using 1 for positive connections (intra-class) and -0.5 for negative ones (cross-class). These parameters are modified in order to observe their influence. Impact of the other parameters is already well discussed in Cosmo et al. (2021) and Feng et al. (2022). Each experiment is repeated with 10 different seeds. Results are shown in Figure 3.



(a) Bar plots of classification accuracy with standard error.



(b) Final weights of the last layer for initialization with $1/-0.5$, $1/0$, and random, respectively. Each one of these models contains 16 graph filters, was using 3 aggregation functions (sum, mean, max, in that order) and was trained on a binary classification task, hence $\mathbf{W} \in \mathbb{R}_{3\cdot 16\times 2}$. Column with red and blue weights, respectively, represents aggregated kernel response which increases logit for class 0 and decreases logit for class 1, hence explicitly describes model's decision process.

Figure 3: Ablation study results.

B.2 Classification accuracy

We assess the performance of our proposed GKConv models across five publicly available graph classification datasets: PROTEINS (Borgwardt et al., 2005), ENZYMES Schomburg et al. (2004) for binary and multi-class classification of biological and chemical compounds, respectively. Moreover, we perform experiments on social datasets: IMDB-BINARY, IMDB-MULTI, and COLLAB (Yanardag & Vishwanathan, 2015). To ensure a fair comparison with state-of-the-art GNNs, we follow the cross-validation procedure outlined in Errica et al. (2022). Employing a 10-fold cross-validation, we follow the identical dataset index splits as described in Errica et al. (2022). Table 1 shows accuracies achieved by our models (DiscGKConvIC with Weisfeiler-Lehman Kernel and DiffGKConvIC with Random Walk Kernel) in comparison to other graph kernel based GNNs (see Errica et al. (2022) for comparison with more GNNs).

Table 1: The mean accuracy and standard deviation.

	PROTEINS	ENZYMES	IMDB-B	IMDB-M	COLLAB
RWGNN	74.7±3.3	57.6±6.3	70.8±4.8	48.8±2.9	71.9±2.5
GKNN	70.9±2.9	29.3±4.3	70.6±5.7	49.9±2.4	65.6±2.2
KerGNN-1	75.8±3.5	62.1±5.5	74.4±4.3	51.6±3.1	70.5±1.6
DiscGKConvIC	69.3+3.7	22.1±6.6	69.8±5.4	47.3±1.7	59.1±2.4
DiffGKConvIC	74.0±4.5	59.0±4.3	71.5±3.7	51.8±2.2	62.8±2.2