MOSE-GNN: A motif-based self-explaining graph neural network for Molecular Property Prediction

Apurva Kokate, and **Xiaoli Z. Fern**
Oregon State University, Corvallis
kokatea@oregonstate.edu

Abstract

Graph Neural Networks (GNNs) have shown significant utility in molecular property prediction but lack interpretability. Most existing interpretability methods focus on instance-based explanations at the node or edge level. Such methods fail to provide a holistic understanding of how key molecular structures influence the model's predictions. This underscores the need for a model-based approach that offers explanations in terms of crucial motifs and their impact on the model's overall decision-making. To address this challenge, we introduce MOtif-based Self-Explaining GNN (MOSE-GNN), an ante-hoc method that integrates motif importance scoring into the GNN architecture. MOSE-GNN assigns global importance scores to predefined motifs, which are shared among instances and generated using RDKit's BRICS Molecular Segmentation function. These scores determine the extent to which the model utilizes information from each motif to predict each class, serving as an explanation for the motif's contributions to the class prediction. Our results on three classification tasks: mutagenicity, blood-brain barrier permeation, and cardiotoxicity demonstrate that MOSE-GNN generates meaningful motif importance scores without sacrificing predictive performance and, in some cases, even improves it.

1 Introduction

Graph Neural Networks (GNNs) have emerged as powerful tools for molecular property prediction as molecules can naturally be represented as graphs[1–3]. As GNNs become more prevalent in high-stakes decision-making areas, particularly in the pharmaceutical and chemical industries, the need for models that predict accurately and also explain their predictions is increasingly pressing[4, 5].

Current methods for explaining GNNs primarily focus on instance-specific explanations, either at the node/edge level [6–9] or the motif level [10, 11]. While these methods can identify influential atoms, bonds, or substructures for individual predictions, they do not offer a broader interpretability of how molecular substructures—specifically functional groups, which are parts of molecules responsible for characteristic reactions [12]—consistently influence model decisions across multiple instances. Recently, some works have extended explanations to the model level for GNNs. Azzolin et al. [13] use PGExplainer [7] to generate instance-level explanations and integrate them into a global, logic-based formula over clusters of local explanations. Other model-level explanation methods [14–16] take a generative approach, training a graph generator to produce patterns that maximize specific predictions of the GNN model. These approaches are post-hoc, developed to explain a model after it has been trained. For post-hoc explainers focused on providing motif-level explanations, there is a risk of interpretative bias if the model itself does not reason at the motif level. In contrast, there is a growing consensus that ante-hoc interpretability [17–19], where interpretability is integrated directly within the model, provides a more robust framework. More recently, Graph Kernel neural networks [20] have emerged as an interesting alternative architecture for graph learning that provides ante-hoc interpretability through learned structural masks. However, a potential limitation of using these learned structure masks as explanations is that they may not correspond to meaningful chemical substructures, which could reduce their interpretability for domain experts.

A. Kokate et al., MOSE-GNN: A motif-based self-explaining graph neural network for Molecular Property Prediction (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

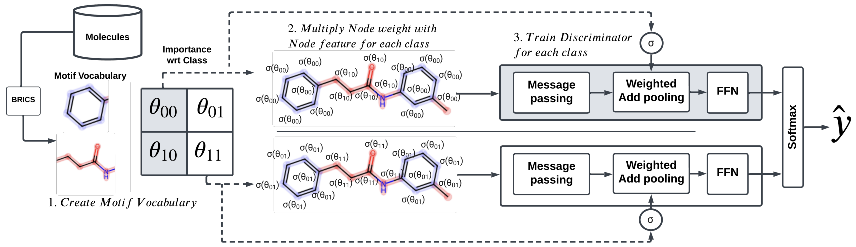

Figure 1: MOSE-GNN: A dual channel GNN where the Nodes features are masked using $\theta_{i,c}$ the importance of motif i for class c.

In this work we propose MOtif-based Self-Explaining GNN (MOSE-GNN), a novel ante-hoc method that can be readily integrated into any message passing graph neural networks to enhance its interpretability by incorporating motif importance scoring directly into the architecture. Unlike traditional instance-based methods, MOSE-GNN assigns global importance scores to predefined motifs shared across all instances in the dataset. These motifs are identified using RDKit's BRICS Molecular Segmentation function[21], a tool designed to break down molecules into meaningful substructures. The normalized importance scores, ranging from 0 to 1, determine how much information from each motif is utilized by the model in predicting molecular properties. By linking motif importance directly to class prediction, MOSE-GNN provides clear, interpretable insights into how molecular substructures contribute to different molecular properties.

We demonstrate the effectiveness of MOSE-GNN through its application to three important molecular classification tasks: mutagenicity, blood-brain barrier permeation, and cardiotoxicity[12]. Testing our method in conjunction with three popular GNN architectures, our results show that MOSE-GNN consistently maintains high predictive performance while generating meaningful motif importance scores, offering a transparent explanation of the model's decision-making process. This approach represents a significant step towards more interpretable and trustworthy GNN-based models in molecular property prediction, addressing a critical gap in the current landscape of GNN interpretability.

2 The proposed method

Figure 1 provides an overview of the proposed MOSE-GNN method. We focus on graph-level classification task (more specifically, binary classification in this work). Given the training set, we first build a vocabulary of motifs that are used as part of the input to the MOSE-GNN architecture.

2.1 Building the motif vocabulary

We create a vocabulary of motifs from the training data by applying RDKIT's[21] BRICS module to fragment the molecules in the training set into functional groups. These fragments are synthetically accessible and follow predefined rules to cleave bonds, which facilitate recombination. Further, unlike fingerprinting and other decomposition methods supported in RDKIT such as RECAP, Murcko Scaffolding and HierS decomposition, BRICS provides us with non-overlapping fragments.

2.2 MOSE-GNN

Given a fixed vocabulary V of motifs $\{m_i\}_{i=1}^{|V|}$, the model learns a multi-channel GNN, where each channel corresponds to a specific class c and is associated with a unique set of motif importance parameters $\Theta_c = \{\theta_{i,c}\}$. Here $\theta_{i,c}$ specifies the importance of motif i for class c. In this framework, each channel of the GNN independently evaluates the input graph based on the motif importance parameters Θ_c for class c. The resulting class-specific scores are then normalized via softmax to generate probabilistic predictions. The motivation behind this design is to allow the model to capture class-specific motif relevance, enabling it to identify motifs as evidence independently for each class and to facilitate class-specific interpretation of motif relevance.

A distinctive feature of our approach is that we identify motifs within the input graph and assign importance parameters directly to nodes in those motifs. For example, in Figure 1, the input graph contains two instances of motif 0 and one of motif 1; nodes in motif 0 receive $\theta_{0,c}$, and nodes in motif 1 receive $\theta_{1,c}$. The graph is then processed using a modified message-passing GNN that incorporates these node-specific weights, allowing motif importance to inform model predictions.

Given the input graph $G = (X, A)$, where X are node features and A is the adjacency matrix, the class-specific GNN channels compute a score for class c as $G_c = FFN(Readout(\Theta_c, MPNN(\Theta_c, X, A)))$. In this formulation, both the message passing function $MPNN$ and the $Readout$ function integrate the motif importance weights Θ_c to generate a representation of the input graph, which is then fed through a feedforward network FNN to produce the score for class c.

Message Passing. One advantage of our method is that it can be readily integrated with any message passing architecture. The following equation describes how the class c importance parameter Θ_c can be incorporated into a generic message passing mechanism, which updates the embedding of a node v by aggregating information from its neighbors denoted by $N(v)$.

$$h^k_{v,c} = UPDATE^k_c\left((w_{v,c})^{I(k\neq 1)} \cdot h^{k-1}_{v,c}, AGG\left(\{MESG(h^{k-1}_{v,c}, h^{k-1}_{v',c}) : v' \in N(v)\}\right)\right) \quad (1)$$

where $UPDATE$, AGG and $MESG$ are architecture specific update, aggregation and message computation functions respectively. Here we use $i(v)$ to denote the motif of node v[1], and compute $w_{v,c} = \sigma(\theta_{i(v),c})$, transforming the importance score of the motif into a weight between 0 and 1.

This formulation differs from the classic GNN formulation by the inclusion of the term $(w_{v,c})^{I(k\neq 1)}$, which takes effect only in the first iteration ($k = 1$). In this initial step, it scales the node's representation by the corresponding motif weight, allowing motif importance to influence the nodes embedding from the start. Note that for zero weight nodes, the GNN will ignore their identity in their representation but still allow them to pass messages from neighboring nodes in subsequent iterations. This design allows the network to capture long-range interactions even when intermediary nodes are deemed unimportant, allowing information to flow through without directly incorporating their features into the representation.

Readout function. In our design, unimportant nodes with $w \approx 0$ serve only as intermediaries for passing messages. To minimize the impact of unimportant nodes on the final representation, we use a weighted readout function: $h_{G,c} = \sum_{v\in G} \frac{w_{v,c}\cdot h^K_{v,c}}{\sum_{u\in G} w_{u,c}}$, ensuring that nodes with low weights have minimal impact on the final graph-level representation.

Unknown motifs. Our motif vocabulary is built from the training data. In testing, we may encounter unseen motifs. Additionally, we filter out rare motifs during training, treating them as unknowns. Nodes belonging to a unknown motif are assigned a default weight value of 1, which remains fixed to avoid learning importance parameters for infrequent motifs, reducing the risk of overfitting. We use a frequency threshold of 3 for Mutagenicity and BBBP and 20 for the hERG dataset to ensure a moderate vocabulary size as well as sufficient node coverage, resulting in 561 motifs for Mutagenicity, 351 for BBBP and 308 for hERG.

2.3 Training objective

We train our model to reduce the cross entropy loss on the target label of the graph and regularize over motif parameters. The total loss is expressed below:

$$L_{CE} + \lambda_1 \sum_{i,c} \sigma(\theta_{i,c}) + \lambda_2 \sum_{i,c} H(\sigma(\theta_{i,c})) \quad (2)$$

where the first term L_{CE} aims to minimize the prediction loss (cross-entropy), the second term seeks to sparsify the importance weights, and the third term $H(\cdot)$ tries to reduce the entropy of the motif parameter to encourage extreme values of 0s or 1s. λ_1 and λ_2 are regularization coefficients and set to 10^{-3} and 0.2 respectively.

3 Experiments

We conduct experiments with three binary classification tasks: Mutagenicity[22, 23], BBBP and HERG [12, 24]. For all three datasets, we used the same train/validation/test splits as [12].

Model configuration and hyperparameters We test our method with GIN[25], GCN[26] and GAT[27] architecture. For all datasets, we use two message passing layers, and a hidden dimension of 16. For training, the batch size is 64 and the learning rate is 0.0001 for GNN parameters and 0.001

[1] In this work, the motifs are non-overlapping, thus a node can only belong to a single motif.

Table 1: Comparing the prediction performance between MOSE-GNN and the vanilla GNN using GIN and GCN architectures for three graph classification datasets.

Dataset	Model	GIN architecture			GCN architecture		
	AUROC	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
HERG	Vanilla	.790 ± .01	.773 ± .01	.752 ± .01	.713 ± .02	.723 ± .01	.693 ± .02
	MOSE-GNN	**.836 ± .01**	**.790 ± .01**	**.785 ± 0.00**	**.807 ± .01**	**.779 ± .01**	**.780 ± .01**
MUTAG	Vanilla	.861 ± .01	.857 ± .01	.834 ± .01	.779 ± .01	.774 ± 0.02	.756 ± .02
	MOSE-GNN	**.900 ± .01**	**.882 ± .01**	**.865 ± .00**	**.873 ± .01**	**.855 ± 0.01**	**.840 ± .012**
BBBP	Vanilla	.893 ± .01	.860 ± .01	**.872 ± .01**	.830 ± .01	.828 ± .01	.826 ± .01
	MOSE-GNN	**.942 ± .01**	**.911 ± .02**	.849 ± .02	**.910 ± .01**	**.889 ± .01**	**.845 ± .01**

for the motif importance parameters. The maximum number of training epochs is set to 200 with early stopping based on validation loss. Atoms are represented using one-hot node features. The bonds are represented as edges in the graph.

Prediction Performance. We measure prediction performance using AUROC. For the baseline, we use a single-channel GNN model with doubled hidden dimensions to match the complexity of our dual-channel GNN. Each experiment is repeated four times with random seeds, and performance is reported in Table 1 which shows MOSE-GNN performs comparably to the vanilla model, outperforming it on Mutagenicity and hERG tasks but slightly underperforming on the BBBP task in the GIN model. The GIN model's additional MLP layer, while improving vanilla model performance, may have increased overfitting and affected interpretability when paired with motif importance scores.

Motif importance. Quantitative evaluation of the motif importance scores is beyond the scope of this abstract, but we provide visualization of the learned motif importance scores in the appendix (Figure 2), which offers several qualitative insights. First, we observe that MOSE-GNN is adept at identifying unimportant motifs containing no information about the property. These are demonstrated as the grey points in the bottom left quadrants of Figure 2 indicating that masking out motifs with the low θ values produced little to no impact on the model's predictions.

We also observe that frequently, motifs with a high θ value for the positive class are indeed positively associated with the property (more red dots in the top half of the figures). However, this is not always the case (see GAT trained on BBBP). This indicates that motifs can serve as negative evidence, suggesting the need to combine the marginal contributions of the motif with the importance score to gain a more complete understanding of the roles of the motifs.

For the Mutagenicity dataset, we observed that the nitrogen-based functional groups, which are known to be informative signatures for mutagencity, have high θ values for both classes, suggesting that these motifs are being used as positive evidence for the mutagen channel while simultaneously acting as negative evidence for the non-mutagen channel.

We also observe that the motif importance scores for GIN tend to be centered around 0.5 for both classes, suggesting that GIN has learned a more complex rationale than can be fully captured by our motif scores, likely due to its higher model complexity.

4 Conclusions and future work

We present MOSE-GNN, a motif-based self-explaining GNN model. Similar to how logistic regression assigns weights to individual features, MOSE-GNN assigns importance scores to motifs—structurally reoccurring subgraphs within molecules. These motif weights indicate the relevance of specific molecular substructures to a given class, offering a global model-based explanation.

Our preliminary results indicate that MOSE-GNN is a promising approach to introduce model-level interpretability into any message passing GNN architecture, while performing on par with traditional GNN models. It shows promising results in filtering out irrelevant motifs and highlighting Key motifs that can serve as both positive and negative evidence for different classes, demonstrating the model's nuanced understanding of molecular structures.

Future work will focus on quantitatively evaluating the motif importance scores and explore their practical implications. We will also refine the motif generation process to further enhance the model's generalization, followed by more extensive experiments considering different prediction tasks including regression problems.

References

[1] Pedro Quesado, Luis H. M. Torres, Bernardete Ribeiro, and Joel P. Arrais. A hybrid gnn approach for improved molecular property prediction. *Journal of Computational Biology*, 0(0): null, 0. doi: 10.1089/cmb.2023.0452. URL https://doi.org/10.1089/cmb.2023.0452. PMID: 39082155. 1

[2] Félix Therrien, Edward H Sargent, and Oleksandr Voznyy. Using gnn property predictors as molecule generators. *arXiv preprint arXiv:2406.03278*, 2024.

[3] Hanxuan Cai, Huimin Zhang, Duancheng Zhao, Jingxing Wu, and Ling Wang. Fp-gnn: a versatile deep learning architecture for enhanced molecular property prediction. *Briefings in bioinformatics*, 23(6):bbac408, 2022. 1

[4] Ignacio Ponzoni, Juan Antonio Páez Prosper, and Nuria E Campillo. Explainable artificial intelligence: A taxonomy and guidelines for its application to drug discovery. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 13(6):e1681, 2023. 1

[5] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures, 2020. URL https://arxiv.org/abs/2002.03244. 1

[6] 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. 1

[7] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020. 1

[8] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, pages 12241–12252. PMLR, 2021.

[9] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM web conference 2022*, pages 1018–1027, 2022. 1

[10] Zhaoning Yu and Hongyang Gao. Motifexplainer: a motif-based graph neural network explainer. *arXiv preprint arXiv:2202.00519*, 2022. 1

[11] Feng Ding, Naiwen Luo, Shuo Yu, Tingting Wang, and Feng Xia. Mega: Explaining graph neural networks with network motifs. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2023. doi: 10.1109/IJCNN54540.2023.10191684. 1

[12] Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, et al. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature Communications*, 14(1):2585, 2023. 1, 2, 3

[13] Steve Azzolin, Antonio Longa, Pietro Barbiero, Pietro Liò, and Andrea Passerini. Global explainability of gnns via logic combination of learned concepts. *arXiv preprint arXiv:2210.07147*, 2022. 1

[14] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 430–438, 2020. 1

[15] Xiaoqi Wang and Han-Wei Shen. Gnninterpreter: A probabilistic generative model-level explanation for graph neural networks, 2024. URL https://arxiv.org/abs/2209.07924.

[16] Zhaoning Yu and Hongyang Gao. Mage: Model-level graph neural networks explanations via motif-based graph generation. *arXiv preprint arXiv:2405.12519*, 2024. 1

[17] Giuseppe Serra and Mathias Niepert. L2xgnn: learning to explain graph neural networks. *Machine Learning*, pages 1–23, 2024. 1

[18] Mert Kosan, Arlei Silva, and Ambuj Singh. Robust ante-hoc graph explainer using bilevel optimization. *arXiv preprint arXiv:2305.15745*, 2023.

[19] Michela Proietti, Alessio Ragno, Biagio La Rosa, Rino Ragno, and Roberto Capobianco. Explainable ai in drug discovery: self-interpretable graph neural network for molecular property prediction using concept whitening. *Machine Learning*, 113(4):2013–2044, 2024. 1

[20] Luca Cosmo, Giorgia Minello, Alessandro Bicciato, Michael M Bronstein, Emanuele Rodolà, Luca Rossi, and Andrea Torsello. Graph kernel neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1

[21] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013. 2

[22] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023. 3

[23] Zhenxing Wu, Dejun Jiang, Jike Wang, Chang-Yu Hsieh, Dongsheng Cao, and Tingjun Hou. Mining toxicity information from large amounts of toxicity data. *Journal of Medicinal Chemistry*, 64(10):6924–6936, 2021. 3

[24] 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, 2018. URL https://arxiv.org/abs/1703.00564. 3

[25] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 3

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 3

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

A Appendix

A.1 Hyperparameter tuning

We did not perform extensive hyper-parameter tuning, except for tuning the learning rating to speed up the training loss reduction as well as optimizing validation performance. Specifically, we searched for the learning rate in {0.01, 0.001, 0.0005, 0.0001}. We fixed the entropy regularization parameter to be 0.2 and adjusted the L1 regularization parameters to ensure that the different loss terms are relatively balanced in their contributions.

A.2 Results for GAT Model

We report the prediction performance and observe that MOSE-GNN outperfoms the Vanilla model for GAT shown in 2

Table 2: Comparing the prediction performance between MOSE-GNN and the vanilla GNN using GAT architecture for three graph classification datasets.

Dataset	Model	Train	Validation	Test
HERG	Vanilla	0.700 ± 0.03	0.705 ± 0.03	0.680 ± 0.02
	Mose-Dual	**0.806 ± 0.02**	**0.778 ± 0.02**	**0.775 ± 0.01**
MUTAG	Vanilla	0.772 ± 0.01	0.765 ± 0.02	0.747 ± 0.01
	MOSE-GNN	**0.878 ± 0.01**	**0.851 ± 0.00**	**0.843 ± 0.01**
BBBP	Vanilla	0.823 ± 0.02	0.830 ± 0.01	0.805 ± 0.02
	MOSE-GNN	**0.917 ± 0.01**	**0.888 ± 0.01**	**0.835 ± 0.00**

A.3 Interpretation of the motif importance scores

We visualize the learned motif importance scores in a scatter plot (Fig. 2), where each point represents a single motif. The coordinates of each point indicate the importance scores for the two respective classes, and the color represents the motif's marginal contribution to the prediction of the positive class.

To compute the marginal contribution of a motif, we consider all graphs containing that motif. For each graph, we measure the change in the logit score differential between the positive and negative classes by masking the motif (i.e., logit score differential of the original graph minus that of the

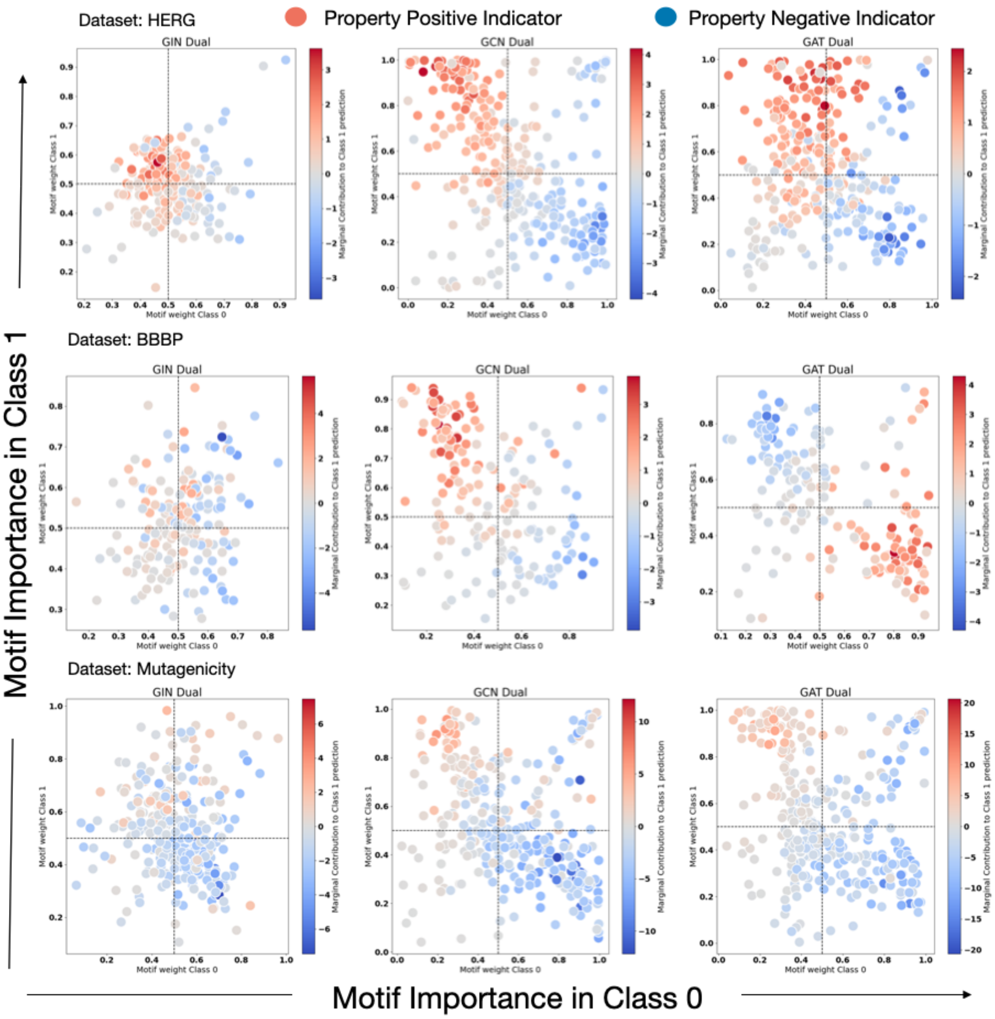

Figure 2: Figure showing learned Motif importance for hERG dataset on first row, BBBP dataset on the second row and Mutagenicity dataset results on third row. Motifs are represented as points on the graph. The x and y axis represent the learned importance weights while the color hue signifies the impact of masking the motif. Top right quadrant indicates motifs are used for both class prediction, top left quadrant indicates the motif is only used for class 1 prediction, bottom left quadrant indicates motif is not used by the model and bottom right quadrant indicates motif is only used for class 0 prediction. A positive score/ red color signifies that removal of the motif decreased models confidence in Class 1, suggesting a positive association between the motif and Class 1 (aka property). Columns 1,2 and 3 indicate GIN, GCN and GAT model performance respectively.

masked graph). A positive change indicates that masking the motif reduces the differential (and thus the probability for the positive property), suggesting a positive association between the motif and the property. We then average this score differential change across all graphs containing the motif.

We note here that the GAT model trained on BBBP learns the opposite rationale (red in the bottom right quadrant and blue in the top left quadrant) than other models. This is because we have a binary classification problem where the channels can either learn to identify posite evidence or negative evidence to get the correct prediction.

A.4 Computation Overhead for MOSE-GNN

The primary computational overhead of MOSE-GNN arises from constructing the Motif Vocabulary. This process involves applying BRICS to fragment each molecular graph in the training set, identifying unique fragments, and applying frequency-based filtering to form the vocabulary. During model training and testing, the multi-channel design increases the memory and computational requirements of the GNN by a factor proportional to the total number of classes.