

APPENDIX

A MORE IMPLEMENTATION DETAILS

A.1 DATASETS

The **ENZYMES** dataset, a collection of protein data obtained from the BRENDA database (Feragen et al., 2013), involves the classification of enzymes into one of six primary EC categories. Detailed statistics of this dataset are presented in Table 1.

The **PROTEINS** dataset, derived from the Dobson and Doig collection (Feragen et al., 2013), consists of protein data with the objective of distinguishing between enzymes and non-enzymes. Table 1 provides detailed statistics of this dataset.

The **D&D** dataset (Dobson & Doig, 2003) comprises high-resolution protein structures taken from a non-redundant selection of the Protein Data Bank. In this dataset, nodes represent amino acids, and an edge is formed between two nodes if they are less than 6 angstroms apart. Detailed statistics of the dataset can be found in Table 1.

The **MUTAG** dataset (Wu et al., 2018) is designed for predicting molecular properties, with nodes representing atoms and edges corresponding to chemical bonds. Each graph carries a binary label that indicates its mutagenic effect. Table 1 displays detailed statistics for the dataset.

The **COLLAB** dataset (Yanardag & Vishwanathan, 2015) focuses on scientific collaborations. In this dataset, each graph represents the ego network of a researcher, with nodes depicting the researcher and their collaborators, and edges signifying collaborations between researchers. The ego network of a researcher can be labeled with one of three categories: High Energy Physics, Condensed Matter Physics, or Astro Physics, reflecting the researcher’s field of study. Detailed statistics of the dataset can be found in Table 1.

The **GraphCycle** dataset (Wang et al., 2024) is a synthetic dataset. Initially, 8~15 Barabási-Albert graphs are generated as communities, each with 10 to 200 nodes. These BA graphs are then interconnected to form two predefined shapes: Cycle and Non-Cycle. Edges between nodes in different communities are randomly added with a probability between 0.05 and 0.15. Detailed statistics of the dataset are given in Table 1.

The **GraphFive** dataset (Wang et al., 2024) is a synthetic dataset. Initially, 8~15 Barabási-Albert graphs are generated as communities, each consisting of 10 to 200 nodes. These BA graphs are subsequently connected in five predefined shapes: Wheel, Grid, Tree, Ladder, and Star. To establish connections between nodes in different clusters, edges are randomly added with a probability between 0.05 and 0.15. Detailed statistics of the dataset can be found in Table 1.

MultipleCycle is a self-designed synthetic dataset. Specifically, we first generate random first-level structures, which consist of either a cycle or a non-cycle structure. For each node in this first-level structure, we further expand it by randomly generating second-level structures, which can either be a cycle or a non-cycle structure. Additionally, each node in the second-level structure is further expanded into one of four third-level structures: a triangle, star, trapezoid, or cycle. The dataset consists of four predefined categories: Pure Cycle, Pure Chain, Hybrid Cycle, and Hybrid Chain, determined based on whether the majority of the nodes at each level form cycle-based or chain-based structures. This hierarchical generation method ensures that each graph exhibits multiple levels of nested structures, with connectivity and patterns varying across the different classes. Specific statistics of the dataset are shown in Table 1.

Table 1: The statistics of real-world datasets.

	#Avg. Nodes	#Avg. Edges	#Classes	#Graphs
ENZYMES	32.63	62.14	6	600
D&D	284.32	715.66	2	1178
PROTEINS	39.06	72.82	2	1113
MUTAG	17.93	19.79	2	188
COLLAB	74.49	2457.78	3	5000
GraphCycle	297.70	697.18	2	2000
GraphFive	375.98	1561.77	5	5000
MultipleGraph	175.33	263.41	4	5000

A.2 BASELINE

To simplify the Tree-like Interpretable Framework (TIF) and investigate the impact of its core components on model performance, we designed a simplified model, named Bi-Tree.

A.2.1 SIMPLIFIED LEARNABLE GRAPH PERTURBATION MODULE

In Bi-Tree, the learnable graph perturbation module from TIF has been simplified to use a set of fixed perturbation terms for each layer. Specifically, while TIF allows each parent node to have independent learnable perturbation matrices, Bi-Tree defines a set of fixed perturbation matrices $P_i^{(l)}$ for each layer l , corresponding to path i . The equation is as follows:

$$S_k^{(l)}(i) = S_k^{(l)} + P_i^{(l)}, \quad i = 1, 2, \dots, M, \quad (1)$$

where $S_k^{(l)}$ represents the clustering assignment matrix generated by the graph coarsening module, and $P_i^{(l)}$ is the fixed perturbation matrix for path i in layer l .

A.2.2 BINARY TREE STRUCTURE WITH LINEAR ROUTERS

Bi-Tree constructs a binary tree structure, where each parent node has only two child nodes. Unlike TIF, which uses multi-level routers, Bi-Tree simplifies each layer’s routers to linear transformations instead of multi-layer perceptrons (MLP). Specifically, the router computes the routing logits $r_k^{(l)}$ based on the node embeddings $Z_{\text{final},k}^{(l)}$:

$$r_k^{(l)} = W_{r,k} \cdot Z_{\text{final},k}^{(l)} + b_{r,k}, \quad (2)$$

where $W_{r,k}$ is the weight matrix for parent node k , and $b_{r,k}$ is the bias term.

A.3 HYPER-PARAMETER SETTINGS

The hyper-parameters used in our framework include batch size, optimizer, learning rate, and epoch. Additionally, several key hyper-parameters control the various loss terms in the model. Specifically, α_1 controls the contribution of the edge prediction loss $\mathcal{L}_{\text{link}}$, which ensures the preservation of graph connectivity during the hierarchical graph coarsening process. α_2 governs the perturbation regularization loss $\mathcal{L}_{\text{perturb}}$, balancing similarity regularization $\mathcal{L}_{\text{similarity}}$ and diversity regularization $\mathcal{L}_{\text{diversity}}$ to ensure the embeddings remain diverse yet close to the original during the learnable graph perturbation module. α_3 adjusts the entropy regularization loss $\mathcal{L}_{\text{entropy}}$, which promotes diverse path selection in the adaptive routing module. The specific settings are provided in Table 2.

Table 2: The statistics of hyper-parameters setting.

	ENZYMES	PROTEINS	D&D	MUTAG	COLLAB	GraphCycle	GraphFive	MultipleGraph
Batch Size	64	64	128	64	64	128	128	128
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.003	0.001	0.001	0.003	0.01	0.01	0.01
Epoch	500	500	500	500	500	500	500	500
α_1/α_2	0.3/0.2	0.3/0.2	0.3/0.2	0.3/0.2	0.3/0.2	0.3/0.2	0.3/0.2	0.3/0.2
α_3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

A.4 MORE DETAILED EXPLANATION OF THE NOTATION

To enhance the readability of the formulas, we will provide a symbol table to further elaborate on the specific meanings of each subscript, offering detailed explanations for each subscript and its function. This will particularly focus on how these subscripts are used in the tree structure model to represent different levels, nodes, and perturbation terms, helping readers better understand our notation system. For details, please refer to Table 3, 4, 5, 6, 7, 8 and 9.

Table 3: Node-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
v_i	i	The i -th node in the graph, representing a specific node.
$\mathbf{Z}^{(l)}$	l	Node embedding matrix after graph convolution at layer l , containing embeddings for all nodes.
$\mathbf{Z}^{(l),k}$	k, l	Node embeddings belonging to node k at layer l , used for representing tree nodes.
$\mathbf{Z}^{(l),k(i)}$	$k(i), l$	Embeddings of node k perturbed by the i -th perturbation at layer l .
$\hat{\mathbf{Z}}^{(l),k}$	k, l	Final aggregated embedding for node k at layer l , used for routing and decisions.

Table 4: Feature and weight-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
\mathbf{X}	None	Input feature matrix, containing the original graph’s node features.
$\mathbf{X}^{(l),k}$	k, l	Feature matrix of node k at layer l , describing its feature state.
$\mathbf{X}^{(l),k(i)}$	$k(i), l$	Feature matrix of node k after applying the i -th perturbation at layer l .
$\mathbf{X}^{(l+1)}$	$l + 1$	Feature matrix of the coarsened graph at layer $l + 1$.
$\mathbf{W}^{(l)}$	l	Weight matrix of the graph convolution at layer l , used for learning graph structural features.
$\mathbf{W}^{(1),r,k}, \mathbf{W}^{(2),r,k}$	r, k	Router weight matrices for node k at layer l , used to compute path selection probabilities.
$\mathbf{b}^{(1),r,k}, \mathbf{b}^{(2),r,k}$	r, k	Bias terms for the router of node k at layer l .

Table 5: Graph structure-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
\mathbf{A}	None	Adjacency matrix of the original graph, representing node connectivity.
$\hat{\mathbf{A}}$	\wedge	Adjacency matrix with self-loops added, improving the stability of graph convolution operations.
$\mathbf{A}^{(l)}$	l	Adjacency matrix of the graph at layer l , describing node connectivity in the coarsened graph.
$\mathbf{A}_{\text{pooled}}^{(l+1),\hat{i}^{l,k}}$	pooled, $\hat{i}^{l,k}, l + 1$	Adjacency matrix of the coarsened graph generated for the selected path $\hat{i}^{l,k}$.

B ADDITIONAL VISUAL EXPLANATIONS

B.1 ADDITIONAL VISUAL EXPLANATIONS FOR THE TREE STRUCTURE

To comprehensively evaluate the interpretability of our proposed TIF, we provide an example that contains the input graph, the root-to-leaf path, the coarsened graphs of each layer, and the final prediction. We conduct a detailed analysis of the multi-granular graph-level nodes and root-to-leaf paths it captures. To facilitate the observation of relationships between structures at different granularities, we visualize our framework’s reasoning process for the MultipleCycle dataset and use different colors to distinguish between various substructures, as illustrated in Figure 1. We observe that TIF effectively captures both local substructures in finer explanations and global structure in coarser explanations, ensuring that key features at different granularities are preserved. The routing module selects the most informative paths through the tree based on multi-granular complexity.

Below, we will take Figure 1 as an example and provide a detailed analysis of the entire process, starting from the input graph, progressing through each intermediate layer and the root-to-leaf path, and finally arriving at the output graph and prediction results and elaborate correlation between the coarsened graph at each layer and the ground-truth.

Firstly, the input graph is a sample from the MultipleCycle dataset, and its category is “Hybrid Cycle”. It corresponds to different ground truths at different levels of granularity. Specifically:

- Its first-level structure is set as a cycle structure based on the ground truth at this granularity level, which determines its cycle attribute.

Table 6: Clustering-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
$\mathbf{S}^{(l)}$	l	Clustering assignment matrix at layer l , representing the probabilities of nodes belonging to different clusters.
$\mathbf{S}^{(l),k}$	k, l	Clustering assignment matrix for node k at layer l .
$\mathbf{S}^{(l),k(i)}$	$k(i), l$	Clustering assignment matrix for node k under the i -th perturbation at layer l .

Table 7: Loss and regularization-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
$\mathcal{L}_{\text{link}}$	link	Edge prediction loss, ensuring connectivity of the adjacency matrix during graph coarsening.
$\mathcal{L}_{\text{similarity}}$	similarity	Similarity regularization, constraining perturbed embeddings to remain close to the original embeddings.
$\mathcal{L}_{\text{diversity}}$	diversity	Diversity regularization, promoting differences between perturbed embeddings.
$\mathcal{L}_{\text{entropy}}$	entropy	Entropy regularization, encouraging diversity in path selection.
\mathcal{L}_{CE}	CE	Cross-entropy loss, optimizing classification objectives.
$\mathcal{L}_{\text{total}}$	total	Total loss function, combining classification, edge prediction, perturbation, and entropy losses.

Table 8: Path and routing-related symbols.

Symbol	Subscript/Superscript	Meaning and Role
$\mathbf{r}^{(l),k}$	k, l	Routing logits for node k at layer l , used to compute path selection probabilities.
$\mathbf{p}^{(l),k,i}$	k, i, l	Path selection probability for node k at layer l , representing the likelihood of selecting branch i .
$\hat{\gamma}_k^{l,k}$	l, k	Optimal path index for node k at layer l , selected based on the maximum probability.
$\text{Path}^{(l),k}$	k, l	Path set at layer l , describing the paths associated with node k .

Table 9: Parameters and hyperparameters.

Symbol	Subscript/Superscript	Meaning and Role
λ_i	i	Weight of the similarity regularization term, controlling the strength of the i -th perturbation.
μ	None	Weight of the diversity regularization term, controlling variation between perturbations.
$\alpha_1, \alpha_2, \alpha_3$	1, 2, 3	Weight coefficients for edge prediction, perturbation, and entropy regularization terms, respectively.
M	None	Number of perturbation branches for each node.
N	None	Number of nodes in the current layer.
$K^{(l)}$	l	Number of clusters at layer l .
L	None	Total number of layers in the tree.

- Its second-level structure is built on the first-level structure, configured as a mixed combination of cycle and non-cycle structures according to the ground truth at this granularity level. The clockwise sequence is cycle, non-cycle, cycle, and cycle, which determines its mixed attribute. (for more detailed information on the dataset, please refer to Appendix A.1.)

Therefore, the final prediction for the input graph in this dataset requires the model to determine:

- whether its first-level granular structure is cycle or non-cycle.
- whether its second-level granular structure represents a mixed combination.

In other words, the model is expected to analyze and make determinations at different granularity levels for this dataset.

Secondly, when the input graph is fed into the model. After passing through a series of graph convolution layers and being processed by the Graph Perturbation Module and Routing Module at the root node of the TIF, the model produces four finer graphs.

We can observe that the finer graphs clearly display the second-level structure of the input graph (in the figures, different colors are used to annotate the nodes of the finer graphs, distinguishing the various second-level structures). From left to right:

- The first finer graph shows a second-level structure starting from the top-left and proceeding clockwise as cycle, non-cycle, cycle, and non-cycle (this structure is not clearly represented).
- The second finer graph shows a second-level structure proceeding clockwise as non-cycle, cycle (which is somewhat ambiguous and not purely cycle), cycle, and cycle.
- The third finer graph shows clockwise as cycle, non-cycle, cycle, and cycle.
- The fourth finer graph shows clockwise as cycle, non-cycle, cycle, and non-cycle.

The model selects the third finer graph, which best reflects the structural information of the input graph. From an interpretability perspective, this layer of finer graphs in the TIF tree model captures the second-level structural information of the input graph. Furthermore, the model selects the finer graph that most effectively represents the second-level structure of the input graph (clockwise: cycle, non-cycle, cycle, cycle). From the perspective of ground truth, the model selects the finer graph that is closest to the ground truth structure and layout of the input graph at this granularity level.

Subsequently, the selected finer graph undergoes another series of graph convolution layers and is processed by the Learnable Graph Perturbation Module and the Adaptive Routing Module at the next layer of the TIF. The model then produces four coarser graphs.

We can observe that the coarser graphs clearly capture the first-level structure of the input graph, which is the cycle structure. To illustrate this correspondence, we have used different colors in the figures to annotate the nodes of the coarser graphs, aligning them with the structures of the finer graphs from the previous layer. From left to right:

- The first coarser graph has two nodes extending outward as small structures from the cycle.
- The second coarser graph has three discontinuous nodes extending outward from the cycle.
- The third coarser graph has two nodes extending outward as small structures from the cycle.
- The fourth coarser graph has three nodes extending outward as small structures from the cycle structure, corresponding to the second-level structure depicted in the finer graph from the previous layer (three cycles organized consecutively).

The model selects the fourth coarser graph, which best represents the structural information of the input graph, as the root node of the TIF. From an interpretability perspective, this layer of coarser graphs in the TIF captures the first-level structural information of the input graph. Additionally, the model selects the coarser graph that not only most effectively represents the first-level structural information of the input graph but also retains the second-level structural information (clockwise: cycle, non-cycle, cycle, cycle, i.e., three cycles organized consecutively). From the perspective of ground truth, the model selects the finer graph that is closest to the ground truth structure and layout of the input graph at this granularity level, while also most accurately preserving the ground truth structural information from the previous granularity level.

Finally, at the root node of the TIF, the prediction is performed, and the model successfully identifies the data as “Hybrid Cycle”. From an interpretability perspective, the TIF effectively captures and explains the key attributes of the MultipleCycle dataset at two distinct granularity levels.

- The second-level granularity characterizes the attributes of being purely cycle, purely non-cycle, or a mixed combination of cycle and non-cycle structures.
- The first-level granularity identifies whether the structure is cycle or non-cycle.

Based on these attributes at the two different granularity levels, the model successfully makes the final prediction for the input graph, completing the classification task.

In addition, the relationship between each coarsened graph and the ground truth lies in the fact that each coarsened graph in the TIF strives to represent the critical structures constructed by the ground truth at the granularity level that the layer aims to explain for the input graph. That is, the coarsened graph obtained at each level by TIF corresponds to the ground truth at that level of granularity.

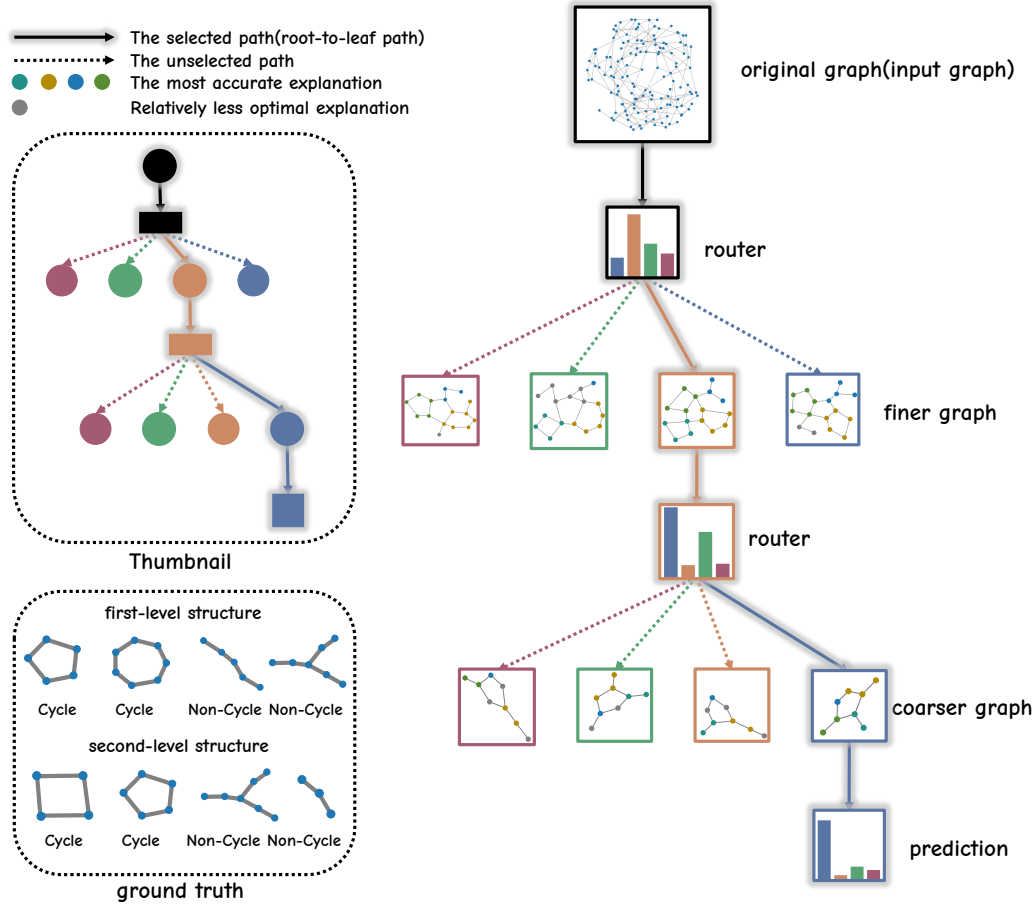


Figure 1: An example which contains the input graph, the root-to-leaf path, the coarsened graphs of each layer, and the final prediction.

B.2 ADDITIONAL VISUAL EXPLANATIONS ON DIFFERENT DATASETS

In this section, we will present additional visualization outcomes of explanations on different datasets. We visualize the explanations generated by our framework on the PROTEINS and D&D datasets. The outcomes are presented in Figure 2 and Figure 3. For clarity of presentation, we only show partial sections of the full explanations for the *finer graph* granularity and *moderate graph* granularity. It can be easily observed that TIF effectively captures both local substructures and global graph patterns, ensuring that key features at different granularities are preserved.

For example, in the PROTEINS dataset, compared to the explanations for non-enzymes, the explanations for enzymes at the *protein molecular* level, or the *coarser graph* granularity, display more long loops and tighter connections. At the *amino acid* level, or the *moderate graph* granularity, enzyme explanations show relatively fixed structural combinations. At the *functional group* level, or the *finer graph* granularity, enzyme explanations reveal denser connections at the active sites.

This observation offers us new insights into differentiating graphs with varying properties, even without specialized knowledge. In the future, we plan to collaborate with domain experts to perform a more thorough analysis.

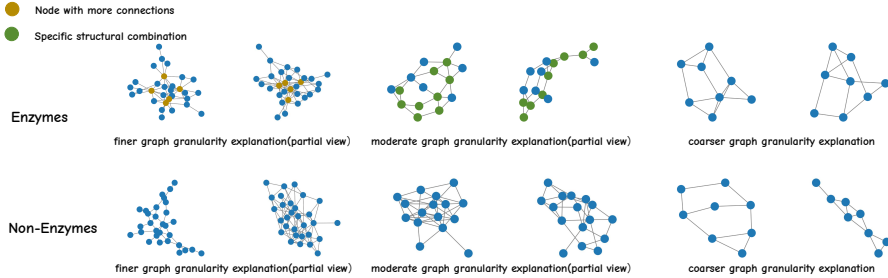


Figure 2: Explanations generated by our framework on the PROTEINS dataset.

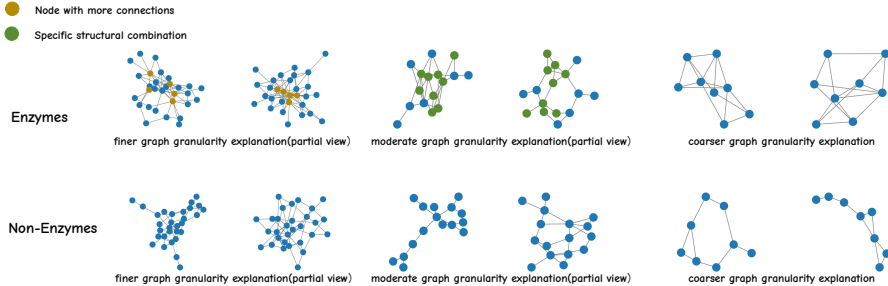


Figure 3: Explanations generated by our framework on the D&D dataset.

B.3 ADDITIONAL VISUAL EXPLANATIONS ON DIFFERENT METHODS

In this section, we observe that TIF effectively captures both local substructures in finer explanations and global graph patterns in coarser explanations, ensuring that key features at different granularities are preserved. The adaptive routing module dynamically selects the most informative paths through the tree based on multi-granular complexity. We also process the same samples using the GIP, GSAT, and ProtGNN and compare the explanations it generates with those produced by our Framework, as illustrated in Figure 4. Our standard for explaining quality is the ability to accurately capture the important features and structural information at each granularity level. Different colors represent structural information learned or captured from the previous level of granularity. Therefore, models like GIP only provide a template based on the entire graph, so the generated explanation is depicted in gray. Compared to those models, TIF’s capability to span from fine-grained local interactions to coarse-grained global structures provides a more transparent and interpretable decision-making process, elucidating how various levels of graph information contribute to final model predictions.

C MORE DETAILED EXPERIMENTAL RESULTS

C.1 PREDICTION PERFORMANCE WITH STD VALUE

To validate the predictive performance of our approach, we compare our framework with widely used GNNs and interpretable GNN models on real-world and synthetic datasets. We apply three independent runs and report the results along with their corresponding std values in Table 10.

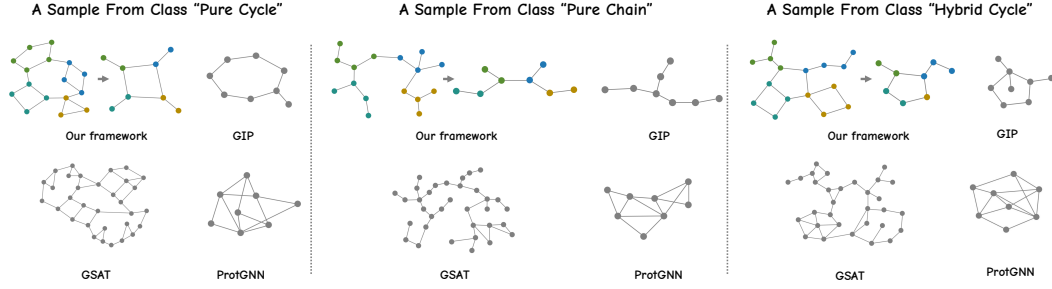


Figure 4: Explanation comparison generated by TIF, GIP, GSAT and ProtGNN on MultipleCycle.

Table 10: Comparison of different methods in terms of classification accuracy (%) and F1 score (%) along with their corresponding standard deviations.

Method	ENZYMES		D&D		PROTEINS		MUTAG		COLLAB		GraphCycle		GraphFive		MultipleCycle	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
GCN	57.23±0.81	51.32±0.33	76.15±2.77	69.12±1.02	78.89±0.90	72.21±3.14	71.82±4.27	63.18±4.36	72.56±1.09	65.78±3.75	79.45±1.04	71.56±1.02	57.37±0.81	53.44±0.55	59.64±4.70	55.56±3.34
DGCNN	59.12±3.30	54.89±1.88	78.23±0.78	71.76±1.59	75.36±1.92	71.43±3.53	58.67±0.80	49.21±0.42	74.88±2.38	68.22±1.44	81.12±2.72	75.34±3.11	57.29±3.33	54.43±2.87	60.71±1.07	56.33±2.41
Diffpool	61.01±2.26	56.98±2.55	81.56±1.31	75.43±4.68	79.52±0.78	78.22±0.87	84.12±2.18	72.45±2.60	72.89±1.39	70.12±1.17	78.34±4.23	71.87±4.59	55.46±1.21	53.57±1.57	56.87±2.03	53.21±2.22
RWNN	54.76±1.43	48.12±3.22	76.89±1.99	74.67±2.18	76.12±1.36	70.89±1.40	88.21±0.21	85.04±0.41	73.45±1.52	68.45±1.97	78.89±1.48	78.76±2.53	56.25±0.42	52.45±1.22	57.16±5.56	54.09±4.10
GraphSAGE	58.12±1.22	44.89±1.32	79.34±5.31	<u>79.23±6.77</u>	79.04±2.15	68.45±2.08	74.23±3.27	71.78±3.62	71.23±1.58	65.45±2.51	77.45±1.49	72.12±2.47	59.11±0.34	52.72±0.36	62.66±0.21	59.34±0.77
ProtGNN	53.21±1.57	43.89±2.36	76.12±1.21	75.23±2.49	76.89±0.52	72.45±1.87	80.34±2.45	61.25±3.83	70.12±0.97	67.89±1.04	80.12±1.21	72.34±2.04	56.38±4.21	54.32±4.37	60.26±3.38	58.41±3.67
KerGNN	55.67±4.22	48.45±2.03	72.89±1.48	68.23±2.36	76.12±2.30	71.12±2.10	71.45±1.08	62.12±1.22	74.12±1.66	69.12±1.97	80.21±0.72	73.89±0.68	58.06±0.11	50.82±1.02	63.22±0.05	57.94±0.33
π -GNN	55.34±0.88	47.12±0.76	79.12±1.10	73.89±1.85	72.34±3.77	68.12±2.21	90.12±0.43	75.12±2.09	73.45±1.52	68.34±3.05	81.45±2.22	76.78±5.62	60.14±0.05	54.07±0.31	64.74±1.21	62.48±1.97
GIB	45.12±3.22	31.67±1.73	77.34±1.69	66.45±0.90	75.12±6.34	70.34±1.05	91.03±4.88	82.12±1.26	73.34±1.79	61.89±1.65	80.67±1.74	74.12±1.98	59.78±0.15	59.24±0.17	63.23±2.63	63.02±2.70
GSAT	61.34±0.65	55.12±1.47	72.12±1.13	67.12±3.22	74.45±0.79	71.89±1.48	<u>94.35±1.12</u>	82.34±1.93	75.87±3.56	63.78±2.59	80.12±0.14	75.08±0.57	59.58±3.09	54.13±2.70	66.49±1.50	65.24±1.53
CAL	61.12±3.24	58.12±4.44	78.12±2.88	68.78±4.76	74.36±4.09	67.12±4.21	89.78±6.99	85.12±8.31	77.12±4.78	64.12±6.25	81.42±2.33	78.12±2.40	56.49±1.44	50.93±2.59	61.77±0.42	58.94±1.73
GIP	60.61±2.41	<u>57.41±2.80</u>	79.32±1.01	75.78±0.36	<u>79.55±0.61</u>	75.28±0.90	91.21±2.25	86.73±2.92	77.49±4.26	67.47±2.11	<u>82.15±1.38</u>	78.31±2.66	60.38±3.33	54.98±1.52	<u>68.72±0.02</u>	<u>66.45±1.34</u>
Ours	58.66±1.44	55.44±2.50	84.19±0.88	81.01±0.76	79.96±0.97	<u>77.21±0.34</u>	94.44±2.44	<u>86.23±3.52</u>	<u>77.29±2.08</u>	67.82±3.27	84.77±0.92	<u>78.49±1.16</u>	64.35±3.55	<u>55.07±2.87</u>	69.04±0.21	67.91±2.77

C.2 EFFICIENCY STUDY

In this section, we analyze the efficiency of the proposed TIF framework and compare its efficiency with several interpretable baselines.

The modular design of TIF ensures efficient computation by progressively reducing the number of nodes through hierarchical coarsening, while controlled perturbations and adaptive routing maintain computational feasibility without compromising model diversity and interpretability.

The running efficiency of the proposed TIF framework is analyzed as follows. In Table 11, we present the time required to complete the training of each interpretable model. The dataset is divided into 10 equal subsets for 10-fold cross-validation, with the time taken by each model being the average of the times required for each fold. Specifically, in each iteration, one fold is held out as the validation set, while the remaining 9 folds are used for training. It should be noted that π -GNN requires an additional pre-training process that takes nearly 72 hours, which significantly impacts its overall computational efficiency. Therefore, the efficiency of π -GNN is considerably lower than our framework. It can be seen that our framework is only slightly less efficient than the KerGNN model and GIP model. Given that our model outperforms KerGNN and GIP in terms of prediction and explanation performance on the vast majority of datasets, as analyzed above, we believe that this slight additional time cost is justified.

C.3 ADDITIONAL ABLATION STUDIES

C.3.1 IMPACT OF THE COMPRESSION RATIO

In this section, we extend the analysis on the impact of the compression ratio q on model performance, conducting experiments across datasets such as MUTAG, and PROTEINS. The results are presented in Figure 5 and Figure 6.

As discussed in the main text, we observe that both classification accuracy and interpretability accuracy tend to decline when the compression ratio is either too high or too low. Specifically, a low compression ratio may introduce noisy structures, thereby hindering the extraction of global information, while a high compression ratio might lead to the loss of critical information.

Table 11: Time consumption of different methods. The table shows the time required (in seconds) to finish training for each interpretable model on various datasets. “*” indicates the method requires an additional pre-training process which takes nearly 72 hours.

Methods	ENZYMES	D&D	COLLAB	MUTAG	GraphCycle	GraphFive
ProtGNN	10245.65s	19312.87s	38021.49s	9239.15s	14396.76s	5022.81s
KerGNN	384.73s	1313.59s	1927.34s	401.34s	198.45s	458.22s
π -GNN*	406.18s	966.94s	1747.55s	462.94s	283.74s	429.82s
GIB	711.57s	2923.67s	4681.74s	3107.31s	1159.82s	1208.78s
GSAT	482.61s	1388.45s	2979.63s	828.19s	568.27s	649.34s
GIP	437.51s	1134.20s	2008.77s	452.26s	235.67s	423.87s
Ours	433.17s	1109.70s	2251.30s	503.18s	359.69s	488.15s

C.3.2 IMPACT OF THE NUMBER OF PATHS

In this section, we present further results on the impact of the number of paths on model performance, covering datasets such as ENZYMES, COLLAB, and FiveGraph shown in Figure 7.

Consistent with the observations in the main text, the experiments reveal that the model achieves the best interpretability when the number of paths is set to four, while performance deteriorates when the number of paths is either too few or too many. Specifically, with only two paths, the model’s choice space is constrained, resulting in insufficient information fusion and an inability to fully leverage the diversity of the graph structure. Conversely, when the number of paths is increased to eight, although potential information channels are expanded, additional noise is introduced, making it challenging for the model to focus on the most critical features. Thus, setting the number of paths to four strikes a balance between information utilization and noise control, effectively improving the model’s interpretability and stability.

C.3.3 IMPACT OF DIFFERENT MECHANISMS

In this section, we further examine routing complexity and perturbation effect by replacing the MLP-based routing module with a simpler linear structure(without the inter-layer adaptive routing mechanism, w/o IAR) and replacing the perturbation module(without the perturbation module, w/o PM). Experiments were conducted across various datasets such as ENZYMES, COLLAB, and GraphFive, with results for classification accuracy and interpretability accuracy presented in Figure 8.

As shown in the figure, the experimental results indicate that the performance is slightly inferior when these mechanisms are used individually, while the combination of these mechanisms achieves the best performance. This superiority stems from the fact that the combination of these mechanisms helps to identify common characteristics in the graph from the perspective of global structure interactions, thereby effectively enhancing the model’s ability to extract global information and interpret key features in complex graph structures.

Specifically, the hierarchical graph coarsening module iteratively aggregates components with similar features or close connections at each layer, forming graph-level representations with higher levels of abstraction. Meanwhile, the graph perturbation module integrates learnable perturbation mechanisms within each lateral layer, resulting in graph-level representations that better reflect the hierarchical structure’s layer-wise characteristics. The combination of these mechanisms is crucial for improving the overall performance of the model.

C.3.4 IMPACT OF LEARNABLE GRAPH PERTURBATION MODULE

In this section, we analyze the impact of the Learnable Graph Perturbation Module on the model and its effectiveness in enhancing diversity. Based on TIF, we created two variants. The first variant replaces the original perturbation terms for each parent node with a set of learnable perturbation terms shared across all parent nodes in each layer(simplified version, SV). The second variant degrades the model by removing the branching structure entirely, effectively eliminating the Learnable Graph Perturbation Module(without the perturbation module, w/o PM).

Experiments were conducted across various datasets, with results for classification accuracy and interpretability accuracy presented in Figure 9.

As shown in the figure, the experimental results indicate that TIF outperforms the other two variants in both classification and interpretability tasks. This suggests that TIF’s perturbation structure effectively captures and learns information that benefits both classification tasks and interpretability.

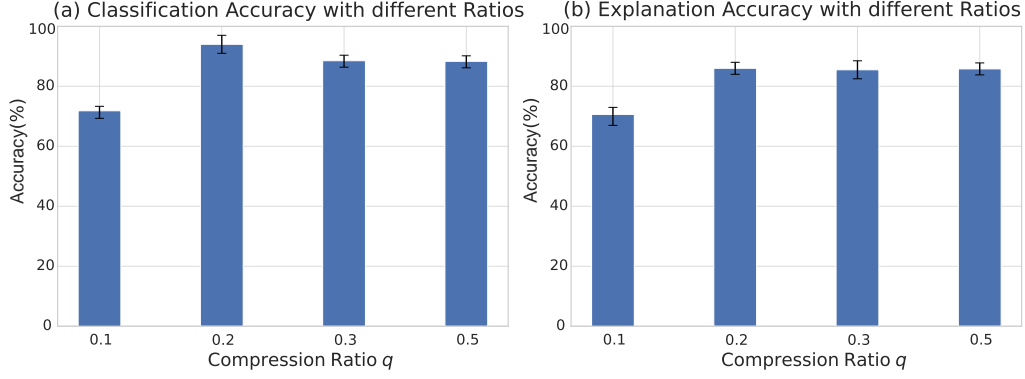


Figure 5: The influence of different compression ratios on the model on the MUTAG dataset.

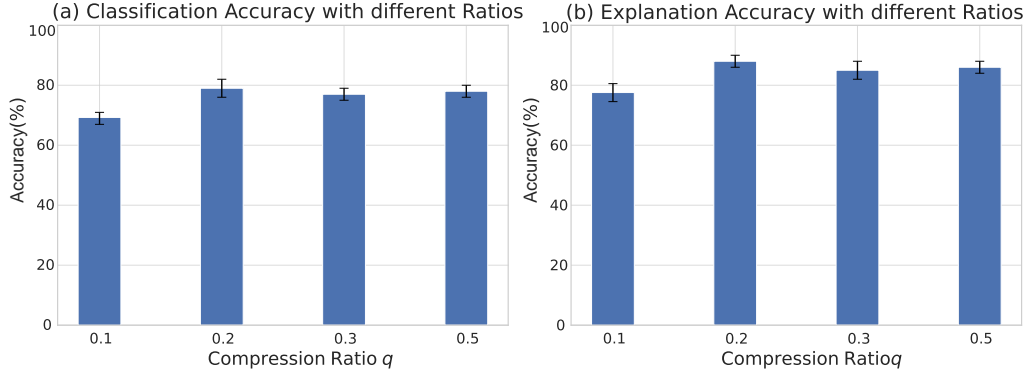


Figure 6: The influence of different compression ratios on the model on the PROTEINS dataset.

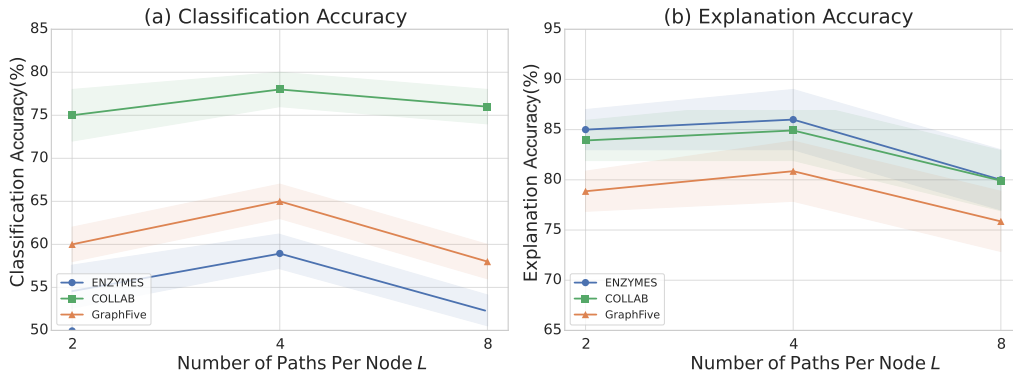


Figure 7: The influence of different numbers of paths per node on the model’s effectiveness.

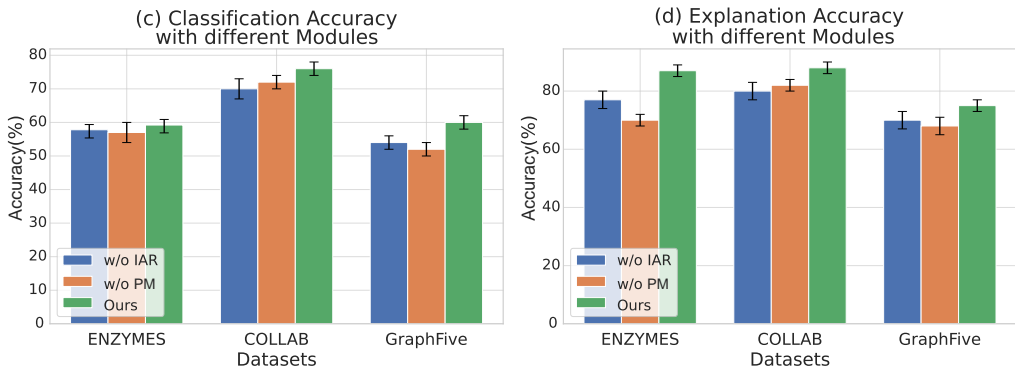


Figure 8: The influence of different modules on the model’s effectiveness.

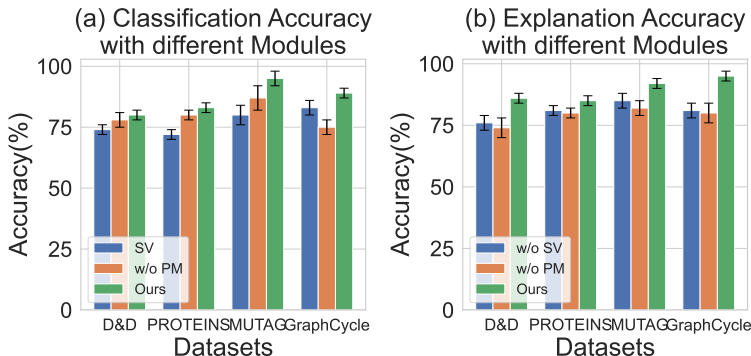


Figure 9: The influence of the learnable graph perturbation module on the model’s effectiveness.

REFERENCES

- Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.
- Aasa Feragen, Niklas Kasenburg, Jens Petersen, Marleen de Bruijne, and Karsten Borgwardt. Scalable kernels for graphs with continuous attributes. In *Annual Conference on Neural Information Processing Systems*, pp. 216–224, 2013.
- Yuwen Wang, Shunyu Liu, Tongya Zheng, Kaixuan Chen, and Mingli Song. Unveiling global interactive patterns across graphs: Towards interpretable graph neural networks. In *ACM Knowledge Discovery and Data Mining*, pp. 3277–3288, 2024.
- 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. *Chemical science*, 9(2):513–530, 2018.
- Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *ACM Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.