Table 3: The best hyperparameter configurations achieved for each dataset using the GIN model.

Dataset	NUM_LAYERS_PRE	NUM_LAYERS	DIM_INNER	DROPOUT	BASE_LR
BZR	1	2	32	0.0	$1e^{-3}$
COX2	2	2	128	0.0	$1e^{-2}$
DD	2	2	128	0.5	$1e^{-3}$
ENZYMES	2	1	128	0.5	$1e^{-3}$
MUTAG	2	1	128	0.0	$1e^{-2}$
NCI1	1	2	128	0.0	$1e^{-3}$
NCI109	2	2	128	0.0	$1e^{-3}$
PROTEINS	1	2	128	0.5	$1e^{-3}$
PTC	1	2	128	0.0	$1e^{-3}$

A TRAINING DETAILS

In this section, we provide detailed information regarding the experimental setup used to obtain the results. We begin by describing the hyperparameters that were cross-validated for the GNN graph-level classifier on GIN. The optimal configuration obtained for each dataset was then used to train the rest of the models. We also present the additional hyperparameters that were cross-validated for the other models apart from GIN. All the fixed hyperparameters can be found in the GitHub repository at `https://github.com/XXXX/XXXX`. For all the experiments, we trained the models using a seed of 42 and report the mean and standard deviation over 5 different dataset splits. We run the models based on reinforcement learning, i.e., SUGAR (`https://github.com/RingBDStack/SUGAR`) and GCIP, for 1000 epochs and the rest of the models for 500 epochs. The experiments were conducted on a single CPU with 10GB RAM.

GIN Hyperparameters The following hyperparameters were cross-validated with the corresponding values:

- NUM_LAYERS_PRE: Number of layers of the MLP used prior to the GNN. Cross-validated values: $[1, 2]$.
- NUM_LAYERS: Number of layers of the GNN. Cross-validated values: $[1, 2]$.
- DIM_INNER: Number of neurons in each (graph) neural network layer. Cross-validated values: $[32, 128]$.
- DROPOUT: Probability of applying dropout. Cross-validated values: $[0.0, 0.5]$.
- BASE_LR: Base learning rate used for training. Cross-validated values: $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$.

Table 3 presents the best configuration obtained for each dataset, which was subsequently used to train the remaining models.

TopK Hyperparameters The following hyperparameter was cross-validated with the corresponding values:

- RATIO: Percentage of nodes to retain. Cross-validated values: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

RNDN Hyperparameters The following hyperparameter was cross-validated with the corresponding values:

- RATIO: Percentage of nodes to retain. Cross-validated values: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

RNDE Hyperparameters The following hyperparameter was cross-validated with the corresponding values:

Table 4: The best hyperparameter configurations achieved for each dataset using the GCIP model for node (left) and edge (right) removal. d represents the maximum desired node/edge ratio.

Dataset	GCIP_N				GCIP_E			
	λ	d	SPLITS	BASE_LR_2	λ	d	SPLITS	BASE_LR_2
BZR	0.50	0.30	[0.5, 0.4, 0.1]	1e-4	0.50	0.30	[0.6, 0.3, 0.1]	1e-4
COX2	0.50	0.30	[0.6, 0.3, 0.1]	1e-4	0.00	0.25	[0.5, 0.4, 0.1]	1e-4
DD	0.00	0.95	[0.6, 0.3, 0.1]	1e-4	0.50	0.30	[0.6, 0.3, 0.1]	1e-4
ENZYMES	1.00	0.30	[0.5, 0.4, 0.1]	1e-4	0.50	0.30	[0.5, 0.4, 0.1]	1e-3
MUTAG	0.50	0.75	[0.5, 0.4, 0.1]	1e-3	1.00	0.30	[0.6, 0.3, 0.1]	5e-3
NCI1	0.50	0.30	[0.6, 0.3, 0.1]	1e-4	0.00	0.50	[0.5, 0.4, 0.1]	1e-4
NCI109	0.00	0.75	[0.6, 0.3, 0.1]	1e-4	0.25	0.30	[0.6, 0.3, 0.1]	1e-4
PROTEINS	1.00	0.30	[0.5, 0.4, 0.1]	5e-3	0.75	0.30	[0.6, 0.3, 0.1]	5e-3
PTC	0.00	0.50	[0.5, 0.4, 0.1]	1e-4	0.00	0.95	[0.6, 0.3, 0.1]	1e-3

- RATIO: Percentage of edges to retain. Cross-validated values: $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

DiffPool Hyperparameters The following hyperparameters were cross-validated with the corresponding values:

- ADDITIONAL_LOSSES: Whether or not to include the additional loss proposed in Ying et al. (2018). Cross-validated values: $\{True, False\}$.
- BASE_LR: Base learning rate used for training. Cross-validated values: $\{1e^{-2}, 1e^{-3}, 1e^{-4}\}$.

SUGAR Hyperparameters The following hyperparameters were cross validated with the corresponding values:

- NEGATIVE_SAMPLING_RATIO: This hyperparameter was cross validated for the self-supervised MI Module and it controlled the amount of negative subgraphs being sampled for contrastive learning. Cross-validated values $\{1, 2, 3, 4, 5\}$
- BETA: This hyperparameter acted as coefficient for the self supervised module in SUGAR. Cross-validated values $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$

GCIP Hyperparameters The following hyperparameters were cross-validated with the corresponding values. The same values were cross-validated for both GCIP_N and GCIP_E.Table 4 presents the best configuration obtained for each dataset for both GCIP_N and GCIP_E.

- BASE_LR_2: Base learning rate used for the outer optimization, which corresponds to the reinforcement learning component. Cross-validated values: $\{1e^{-3}, 1e^{-4}\}$.
- SPLITS: The split percentages used for training, validation, and test sets. The test size was kept fixed, while the training and validation sizes were varied. These splits were used for training the base graph classifier and the reinforcement learning component, respectively. Cross-validated values: $\{[0.5, 0.4, 0.1], [0.6, 0.3, 0.1]\}$.

A.1 EVOLUTION OVER EPOCHS

We present (in Figure 5) the evolution of several metrics during the training of the nine datasets discussed in Section 5. Specifically, we analyze the accuracies, PPO rewards, node/edge ratio.

Upon observing the trends across all metrics, we find a consistent and stable evolution throughout the epochs, providing compelling evidence for the effectiveness of the bi-level optimization approach. However, it is important to note that three datasets (NCI1, NCI109, and PTC) do not achieve convergence by epoch 1000. This suggests that extending the training duration may further improve the performance of our proposed method GCIP. Nevertheless, to ensure a fair comparison, we did not pursue this approach.

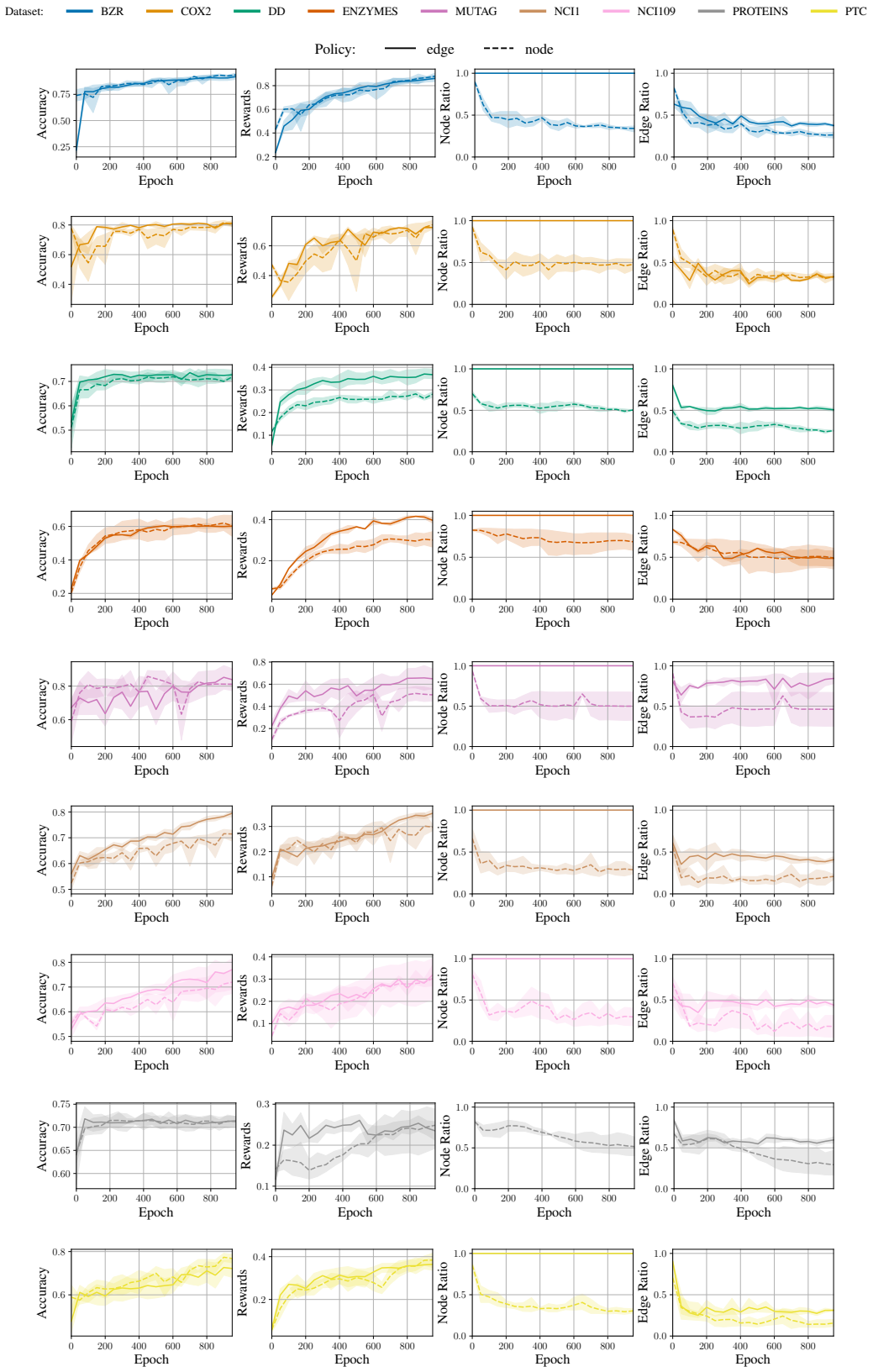


Figure 5: Evolution of different metrics during training for the nine datasets under study.

B GCIP TRAINING PROCEDURE

In this section we include a description of the training procedure of GCIP, summarized in Algorithm 1.

Algorithm 1 Training of GCIP.

Input: Training $\mathcal{D}^{tr}$ and validation $\mathcal{D}^{val}$ sets; Initial graph classifier parameters θ and policy parameters ϕ; Empty buffer $\mathcal{B}$; Learning rates α and β; Number of PPO updates K
while not convergence **do**
  for $\boldsymbol{y}, \mathcal{G} \in \mathcal{D}^{tr}$ **do** ▷ Get a (batch of) sample(s) from the training set
    $\boldsymbol{a} \sim \pi_\phi(\boldsymbol{a}|\mathcal{G})$ ▷ Sample the nodes/edges to be removed using the policy
    $\mathcal{G}_s \leftarrow \mathcal{G}$ ▷ Get the subgraph
    $\hat{\boldsymbol{y}}_s = \mathrm{f}_\theta(\mathcal{G}_s)$ ▷ Get the prediction
    $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{perf}}(\theta, \phi)$ ▷ Update the parameters of the graph classifier
  end for
  for $\boldsymbol{y}, \mathcal{G} \in \mathcal{D}^{val}$ **do** ▷ Get a (batch of) sample(s) from the validation set
    $\mathcal{G}_s \leftarrow \pi_\phi$ ▷ Use the π to sample a sparse graph
    $\hat{\boldsymbol{y}}_s = \mathrm{f}(\mathcal{G}_s)$ ▷ Get the prediction of the sparse graph
    Add tuple $\{\boldsymbol{y}, \mathcal{G}, \hat{\boldsymbol{y}}, \mathcal{G}_s\}$ to the buffer $\mathcal{B}$
  end for
  Compute rewards and prepare mini-batches from buffer $\mathcal{B}$ for PPO updates
  for i = 1, 2, ..., K **do** ▷ Update PPO K times
    $\{\boldsymbol{y}, \mathcal{G}, \hat{\boldsymbol{y}}, \mathcal{G}_s, r\} \sim \mathcal{B}$ ▷ Sample a batch of tuples from the buffer
    Compute advantage estimates $\hat{A}$
    $\phi \leftarrow \phi - \beta \nabla_\phi \mathcal{L}_{\text{spa}}(\theta, \phi)$ ▷ Update policy by maximizing Equation (1)
  end for
  Check for convergence, update convergence flag
end while
Output: Optimal policy parameters ϕ and graph classifier parameters θ

C COMPLETE RESULTS ON ABLATION STUDY

In this section, we provide a complete ablation study on λ and desired node/edge ratio d for all the datasets namely BZR, COX2, DD, ENZYMES, MUTAG, NCI1, NCI109, PROTEINS and PTC.

C.1 ABLATION STUDY ON λ

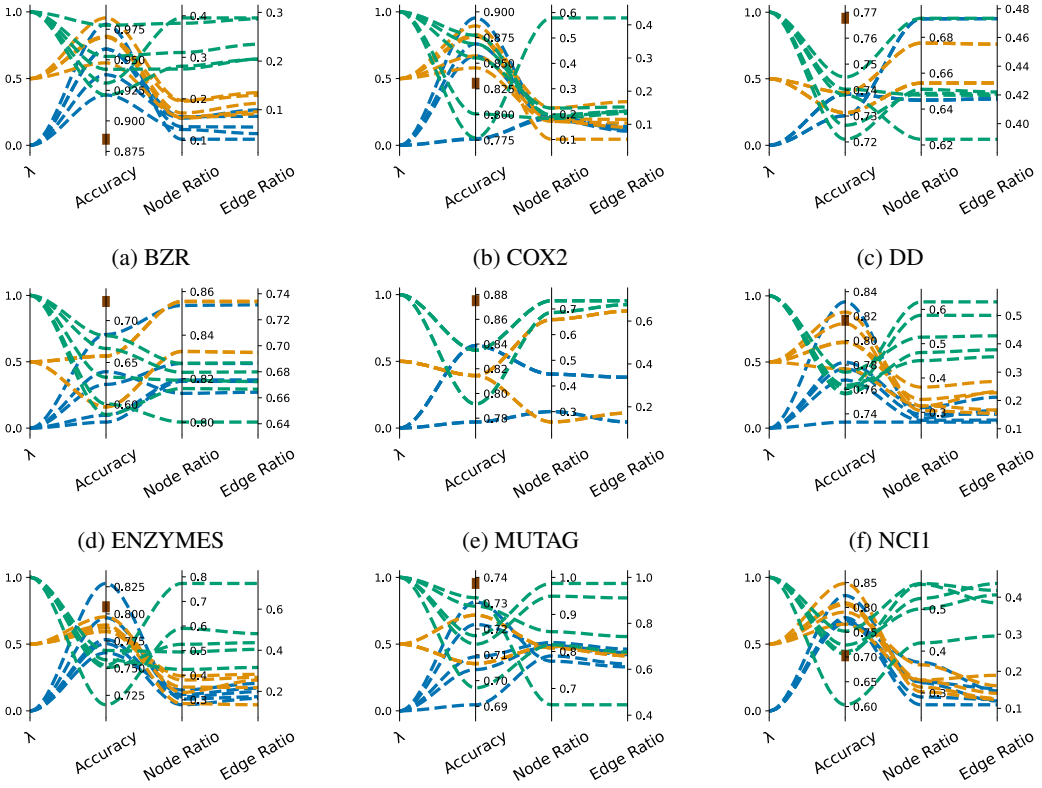
Here we analyze the results of how the parameter λ affects the sparsity and performance of GCIP_E and GCIP_N, on all datasets. Results are shown in Figure 6 for GCIP_N and Figure 7 for GCIP_E.

We can observe that the λ is able to control sparsity in nodes and edges without dropping the accuracy. In general, lower values of λ result in sparser graphs. The effect is really clear for most datasets like NCI1, with the node ratio in Figure 6 ranging from approximately 0.5 to 0.15, and the edge ratio in Figure 7 ranging from approximately 0.8 to 0.4. However, for other datasets like DD, the differences are much smaller, with the node ratio in Figure 6 ranging from approximately 0.48 to 0.4, and the edge ratio in Figure 7 ranging from approximately 0.58 to 0.5.

C.2 ABLATION STUDY ON MAXIMUM DESIRED RATIO d

Here we analyze the results of how the parameter d affects the sparsity and performance of GCIP_E and GCIP_N, on all datasets. Results are shown in Figure 8 for GCIP_N and Figure 9 for GCIP_E.

Similar to the results shown in Figure 6 and Figure 7, we can observe that the parameter d is able to control sparsity in nodes and edges without dropping the accuracy, extensively. Some datasets, such as DD, exhibit more subtle differences. However, in general, lower values of d clearly lead to sparser graphs. This is evident in datasets like NCI1, NCI109, and PTC, where the graph becomes significantly sparser as d decreases.

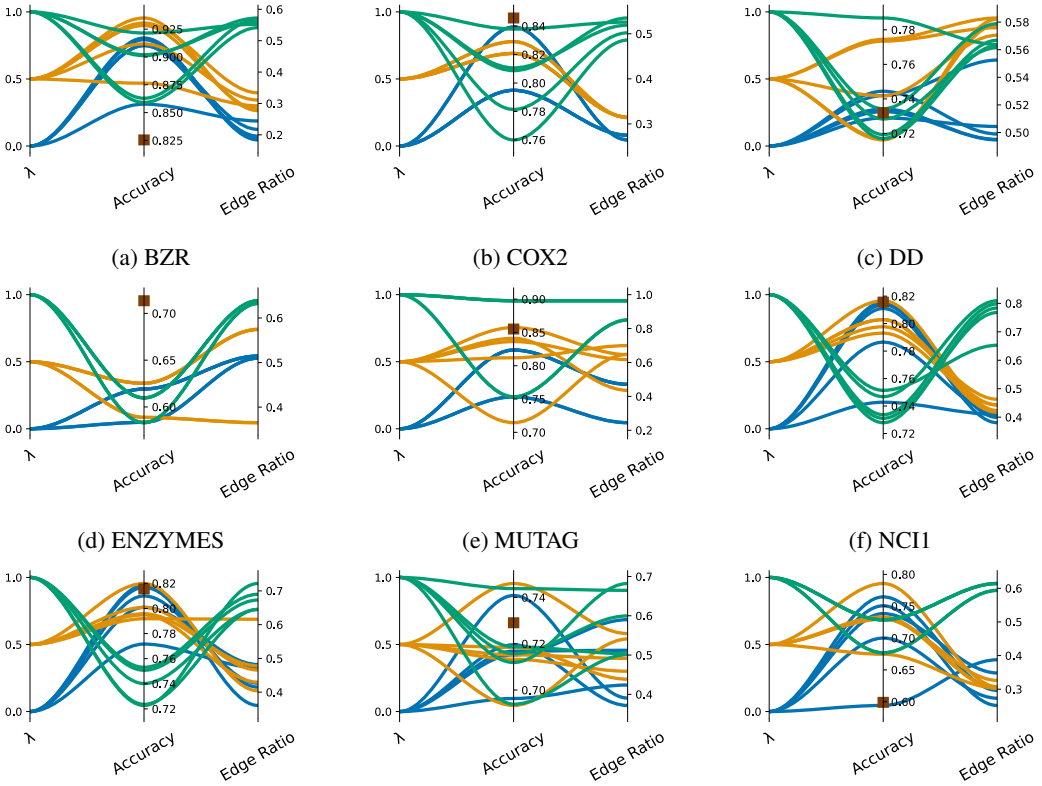


(a) BZR (b) COX2 (c) DD

(d) ENZYMES (e) MUTAG (f) NCI1

(g) NCI109 (h) PROTEINS (i) PTC

Figure 6: Ablation study on λ for GCIP_N, controlling the importance given to performance or sparsity. The brown solid rectangle on the accuracy axis represents the baseline GIN (baseline) accuracy.

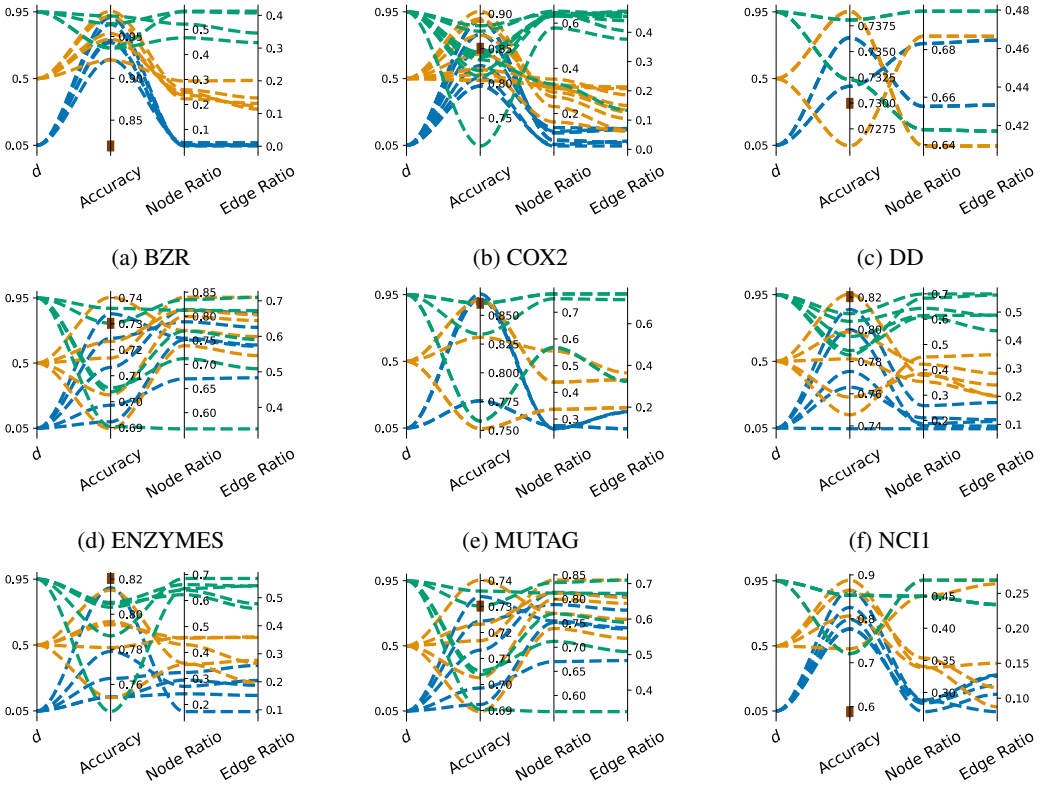


(a) BZR (b) COX2 (c) DD

(d) ENZYMES (e) MUTAG (f) NCI1

(g) NCI109 (h) PROTEINS (i) PTC

Figure 7: Ablation study on λ for GCIP_{E}, controlling the importance given to performance or sparsity. The brown solid rectangle on the accuracy axis represents the baseline GIN (baseline) accuracy.

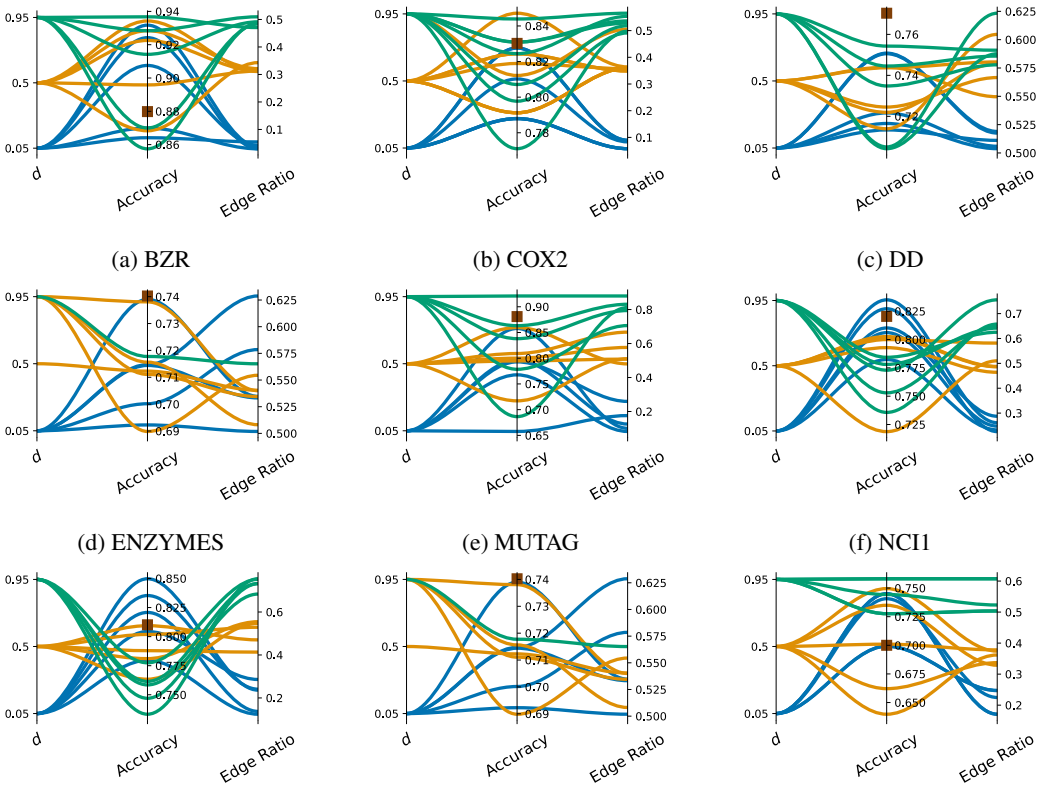


(a) BZR (b) COX2 (c) DD

(d) ENZYMES (e) MUTAG (f) NCI1

(g) NCI109 (h) PROTEINS (i) PTC

Figure 8: Ablation study on d for GCIP$_{\text{N}}$, controlling the importance given to performance or sparsity. The brown solid rectangle on the accuracy axis represents the baseline GIN (baseline) accuracy.



(a) BZR (b) COX2 (c) DD

(d) ENZYMES (e) MUTAG (f) NCI1

(g) NCI109 (h) PROTEINS (i) PTC

Figure 9: Ablation study on d for GCIP$_{\mathrm{E}}$, controlling the importance given to performance or sparsity. The brown solid rectangle on the accuracy axis represents the baseline GIN (baseline) accuracy.

Table 5: Test set accuracy results. We show the mean over 5 independent runs and the standard deviation as the subindex. The last row includes the average ranking of the model across datasets. Best performing models on average are indicated in bold.

Dataset	Full Models		Sparse Models			
	GIN	GCN	$\text{GCIP}_{\text{GIN-N}}$	$\text{GCIP}_{\text{GIN-E}}$	$\text{GCIP}_{\text{GCN-N}}$	$\text{GCIP}_{\text{GCN-E}}$
BZR	81.95 $_{2.78}$	**85.37 $_{2.44}$**	79.90 $_{4.29}$	82.39 $_{5.07}$	79.02 $_{3.22}$	79.54 $_{8.85}$
COX2	**84.58 $_{4.56}$**	78.75 $_{4.52}$	75.49 $_{6.33}$	83.59 $_{2.82}$	77.81 $_{5.03}$	79.05 $_{3.48}$
DD	73.11 $_{2.45}$	**76.97 $_{2.19}$**	75.51 $_{1.86}$	74.40 $_{2.12}$	68.35 $_{2.75}$	70.02 $_{3.92}$
ENZYMES	72.33 $_{4.35}$	**72.67 $_{5.48}$**	63.70 $_{3.89}$	49.53 $_{5.29}$	50.88 $_{6.88}$	56.02 $_{7.60}$
MUTAG	**86.00 $_{4.18}$**	79.00 $_{7.42}$	74.63 $_{2.12}$	74.32 $_{6.34}$	69.89 $_{4.91}$	71.59 $_{6.08}$
NCI1	82.04 $_{1.26}$	**82.29 $_{1.05}$**	73.66 $_{2.74}$	73.51 $_{1.02}$	68.48 $_{2.03}$	69.69 $_{1.47}$
NCI109	**81.59 $_{1.91}$**	79.90 $_{1.79}$	71.88 $_{3.94}$	72.48 $_{2.78}$	64.74 $_{2.71}$	66.93 $_{2.10}$
PROTEINS	72.86 $_{4.22}$	71.19 $_{2.30}$	73.49 $_{7.27}$	**73.50 $_{5.03}$**	69.18 $_{3.39}$	70.23 $_{3.09}$
PTC	59.44 $_{8.91}$	55.00 $_{3.62}$	52.92 $_{8.67}$	**66.06 $_{4.31}$**	61.14 $_{9.79}$	60.00 $_{10.41}$
Accuracy rank	**2.33**	2.33	3.67	2.89	5.33	4.44

Table 6: Test set accuracy results. We show the mean over 5 independent runs and the standard deviation as the subindex. The last row includes the average ranking of the model across datasets. Best performing models on average are indicated in bold.

Dataset	Sparse Models			
	RandomNode	RandomEdge	GCIP_{N}	GCIP_{E}
BZR	64.39 $_{5.62}$	71.71 $_{2.09}$	79.90 $_{4.29}$	**82.39 $_{5.07}$**
COX2	70.42 $_{8.51}$	78.75 $_{9.40}$	75.49 $_{6.33}$	**83.59 $_{2.82}$**
DD	39.66 $_{7.64}$	41.68 $_{3.71}$	**75.51 $_{1.86}$**	74.40 $_{2.12}$
ENZYMES	27.67 $_{6.30}$	29.00 $_{7.23}$	**63.70 $_{3.89}$**	49.53 $_{5.29}$
MUTAG	**75.00 $_{7.07}$**	69.00 $_{17.57}$	74.63 $_{2.12}$	74.32 $_{6.34}$
NCI1	70.07 $_{2.31}$	70.36 $_{3.83}$	**73.66 $_{2.74}$**	73.51 $_{1.02}$
NCI109	68.12 $_{1.70}$	67.63 $_{2.15}$	71.88 $_{3.94}$	**72.48 $_{2.78}$**
PROTEINS	71.01 $_{5.13}$	71.19 $_{3.55}$	73.49 $_{7.27}$	**73.50 $_{5.03}$**
PTC	-	-	52.92 $_{8.67}$	**66.06 $_{4.31}$**
Accuracy rank	3.50	3.12	1.78	**1.56**

C.3 Comparison with different base architecture

Here, we analyze the results of how switching the base model affects the performance. Table 5 summarizes the results for two base architectures GCN and GIN. It can be observed that even when the base architecture changes our models GCIP_{N} and GCIP_{E} achieve considerable accuracy compared to the respective baseline.

C.4 Comparison with random sparser graphs

In this section we compare GCIP with a baseline sparsity approach: we obtain a sparser graph by randomly removing a fixed ratio of nodes or edges of each graph in the dataset. We denote the resulting approaches RandomNode and RandomEdge respectively. We cross-validate the ratio of nodes and edges kept and we used GIN as the base model.

Table 6 shows the accuracy results, and Table 7 shows the results of sparsity. In Table 6, we can observe that GCIP_{N} and GCIP_{E} perform much better in accuracy compared to the RandomNode and RandomEdge models. Additionally, in Table 7, we show that GCIP_{N} achieves the highest Node and Edge rank.

Table 7: Node/Edge ratio (shown in %) for the graphs in the test set for *Sparse* models on 9 different datasets. Numbers in parentheses indicate the ranking of the model for each dataset. The last two rows indicate the average ranking of the models across datasets in terms of node and edge sparsity, respectively. Best performing models on average are indicated in bold.

Dataset	**Nodes/Edges %**	**Sparse Models**				
		RandomNode	**RandomEgde**	**TopK$_{\text{hard}}$**	**GCIP$_{\text{N}}$**	**GCIP$_{\text{E}}$**
BZR	Node Ratio (%)	**18.84 ± 0.13 (1.00)**	100.00 ± 0.00 (4.00)	31.28 ± 0.04 (3.00)	18.87 ± 1.67 (2.00)	100.00 ± 0.00
	Edge Ratio (%)	**2.99 ± 0.25 (1.00)**	9.45 ± 0.04 (2.00)	22.09 ± 1.86 (4.00)	12.80 ± 1.37 (3.00)	29.07 ± 2.89 (5.00)
COX2	Node Ratio (%)	78.89 ± 0.09 (3.00)	100.00 ± 0.00 (4.00)	50.63 ± 0.10 (2.00)	**17.48 ± 1.72 (1.00)**	100.00 ± 0.00
	Edge Ratio (%)	62.00 ± 1.50 (5.00)	39.58 ± 0.05 (3.00)	44.50 ± 3.68 (4.00)	**11.52 ± 2.82 (1.00)**	20.41 ± 1.33 (2.00)
DD	Node Ratio (%)	**9.75 ± 0.02 (1.00)**	100.00 ± 0.00 (4.00)	90.25 ± 0.02 (3.00)	66.64 ± 2.73 (2.00)	100.00 ± 0.00
	Edge Ratio (%)	**0.88 ± 0.03 (1.00)**	69.95 ± 0.00 (4.00)	81.19 ± 0.34 (5.00)	44.20 ± 3.40 (2.00)	58.14 ± 0.77 (3.00)
ENZYMES	Node Ratio (%)	**78.34 ± 0.27 (1.00)**	100.00 ± 0.00 (4.00)	91.74 ± 0.08 (3.00)	81.84 ± 1.76 (2.00)	100.00 ± 0.00
	Edge Ratio (%)	60.36 ± 0.52 (3.00)	**39.45 ± 0.14 (1.00)**	84.25 ± 0.63 (5.00)	67.16 ± 2.88 (4.00)	47.16 ± 12.87 (2.00)
MUTAG	Node Ratio (%)	86.88 ± 0.73 (3.00)	100.00 ± 0.00 (4.00)	73.04 ± 0.89 (2.00)	**66.00 ± 9.87 (1.00)**	100.00 ± 0.00
	Edge Ratio (%)	74.34 ± 2.03 (3.00)	78.95 ± 0.16 (4.00)	**55.00 ± 1.65 (1.00)**	62.15 ± 14.50 (2.00)	95.42 ± 6.27 (5.00)
NCI1	Node Ratio (%)	88.27 ± 0.09 (3.00)	100.00 ± 0.00 (4.00)	71.78 ± 0.05 (2.00)	**32.08 ± 2.77 (1.00)**	100.00 ± 0.00
	Edge Ratio (%)	77.45 ± 0.23 (4.00)	89.25 ± 0.02 (5.00)	48.82 ± 0.52 (3.00)	**19.89 ± 5.17 (1.00)**	25.69 ± 2.02 (2.00)
NCI109	Node Ratio (%)	88.22 ± 0.03 (2.00)	100.00 ± 0.00 (4.00)	91.74 ± 0.06 (3.00)	**53.88 ± 2.06 (1.00)**	100.00 ± 0.00
	Edge Ratio (%)	77.54 ± 0.15 (3.00)	79.24 ± 0.04 (4.00)	89.27 ± 0.33 (5.00)	41.71 ± 4.09 (2.00)	**40.65 ± 5.76 (1.00)**
PROTEINS	Node Ratio (%)	**67.55 ± 0.14 (1.00)**	100.00 ± 0.00 (4.00)	92.27 ± 0.24 (3.00)	85.69 ± 14.06 (2.00)	100.00 ± 0.00
	Edge Ratio (%)	44.30 ± 0.18 (2.00)	**9.38 ± 0.08 (1.00)**	85.76 ± 0.41 (5.00)	75.63 ± 22.91 (4.00)	47.60 ± 8.10 (3.00)
PTC	Node Ratio (%)	-	-	94.90 ± 1.08 (2.00)	**39.16 ± 7.52 (1.00)**	100.00 ± 0.00
	Edge Ratio (%)	-	-	89.94 ± 1.95 (3.00)	**18.06 ± 8.06 (1.00)**	52.30 ± 1.68 (2.00)
Average Rank	Node Rank	1.88	4.00	2.56	**1.44**	-
	Edge Rank	2.75	3.00	3.89	**2.22**	2.78

Table 8: The table below shows the average training time along with the standard deviation of each model on each dataset for one epoch with a batch size of one.

Dataset	Full Models		Sparse Models			
	GIN	DiffPool	SUGAR	$\text{TopK}_{\text{hard}}$	GCIP_{N}	GCIP_{E}
BZR	2.64 $_{0.18}$	4.23 $_{0.12}$	-	3.44 $_{0.27}$	186.96 $_{61.70}$	121.94 $_{2.00}$
COX2	3.07 $_{0.15}$	4.83 $_{0.17}$	-	10.00 $_{2.70}$	411.93 $_{85.97}$	203.93 $_{1.86}$
DD	6.13 $_{0.13}$	-	-	33.31 $_{4.90}$	1848.21 $_{1.12}$	1857.59 $_{1.13}$
ENZYMES	2.40 $_{0.04}$	6.26 $_{0.31}$	132.86 $_{3.77}$	6.08 $_{0.30}$	233.78 $_{6.65}$	287.95 $_{2.20}$
MUTAG	1.09 $_{0.03}$	1.63 $_{0.04}$	47.82 $_{3.45}$	1.71 $_{0.19}$	24.97 $_{1.55}$	22.94 $_{0.30}$
NCI1	15.44 $_{0.19}$	41.74 $_{3.03}$	1143.85 $_{45.27}$	48.05 $_{8.12}$	1596.95 $_{107.83}$	1826.41 $_{51.06}$
NCI109	16.68 $_{1.15}$	43.73 $_{0.44}$	1195.10 $_{47.42}$	43.46 $_{3.09}$	1533.59 $_{15.51}$	1756.84 $_{37.12}$
PROTEINS	5.80 $_{0.23}$	11.79 $_{1.19}$	511.21 $_{22.63}$	13.58 $_{2.58}$	484.97 $_{14.74}$	505.89 $_{2.70}$
PTC	1.68 $_{0.16}$	3.00 $_{0.07}$	95.78 $_{1.47}$	3.28 $_{0.18}$	64.60 $_{1.23}$	67.19 $_{0.32}$

D TIME COMPLEXITY

Here, we analyze the training and inference times of all models on each dataset. We run each model on every dataset for a single epoch, utilizing a batch size of one. To ensure statistical significance and reliability in our measurements, we repeat each experiment ten times. This repetition enables us to calculate the average training and inference times accurately. Such a meticulous approach guarantees robust and precise performance assessment for our models across the diverse set of datasets under examination.

Table 8 and Table 9 show the complete results for the training and inferences times, respectively. If we focus on Table 8, we observe the models that only rely on a single forward pass are considerably faster than the models that rely on a reinforcement learning-based approach. Still, we observe that GCIP achieves comparable speed as SUGAR, and sometimes it is faster, e.g., with PTC or MUTAG. In terms of inference time, in Table 9, we can observe that even tho GCIP is still slower than the *Full Models*, it is considerably faster than SUGAR.

Table 9: The table below shows the average inference time along with the standard deviation of each model on each dataset.

Dataset	Full Models		Sparse Models			
	GIN	DiffPool	SUGAR	$\text{TopK}_{\text{hard}}$	GCIP_{N}	GCIP_{E}
BZR	0.0025 $_{0.0001}$	0.1447 $_{0.0048}$	-	0.0045 $_{0.0003}$	0.0540 $_{0.0257}$	0.0425 $_{0.0011}$
COX2	0.0026 $_{0.0001}$	0.1550 $_{0.0093}$	-	0.0062 $_{0.0008}$	0.1031 $_{0.0349}$	0.0526 $_{0.0014}$
DD	0.0028 $_{0.0004}$	-	-	0.0085 $_{0.0006}$	0.2262 $_{0.0842}$	0.1206 $_{0.0016}$
ENZYMES	0.0024 $_{0.0000}$	0.1992 $_{0.0080}$	0.1399 $_{0.0094}$	0.0063 $_{0.0007}$	0.0496 $_{0.0014}$	0.0625 $_{0.0019}$
MUTAG	0.0029 $_{0.0001}$	0.0596 $_{0.0034}$	0.0135 $_{0.0036}$	0.0065 $_{0.0017}$	0.0448 $_{0.0056}$	0.0415 $_{0.0012}$
NCI1	0.0022 $_{0.0001}$	1.3756 $_{0.1246}$	1.0187 $_{0.1040}$	0.0058 $_{0.0004}$	0.0530 $_{0.0026}$	0.1301 $_{0.0602}$
NCI109	0.0023 $_{0.0001}$	1.3685 $_{0.0988}$	0.9477 $_{0.0132}$	0.0055 $_{0.0002}$	0.0500 $_{0.0006}$	0.0563 $_{0.0009}$
PROTEINS	0.0025 $_{0.0001}$	0.3835 $_{0.0619}$	3.9375 $_{0.0347}$	0.0058 $_{0.0007}$	0.0437 $_{0.0016}$	0.0439 $_{0.0008}$
PTC	0.0028 $_{0.0001}$	0.1076 $_{0.0050}$	0.0398 $_{0.0099}$	0.0058 $_{0.0005}$	0.0395 $_{0.0013}$	0.0418 $_{0.0005}$