

## A EXPERIMENT DETAILS

### A.1 DETAILS ON GRAPH DATASETS

We show the statistics of the datasets for node classification in Table 4. For each dataset, we list its graph statistics, data split, and homophily score, which is computed as follows.

**Homophily.** We adopt the homophily indicator  $H(\mathcal{G})$  of the graph  $\mathcal{G}$  from Liu et al. (2021b), which can be calculated as:

$$H(\mathcal{G}) = \frac{1}{|V|} \sum_{v \in V} \frac{|\{u : u \in \mathcal{N}(v) \text{ and } y(u) = y(v)\}|}{|\mathcal{N}(v)|},$$

where  $|\{u : u \in \mathcal{N}(v) \text{ and } y(u) = y(v)\}|$  denotes the number of  $v'$  s directly connected nodes who have the same label as  $v$  and  $|\mathcal{N}(v)|$  is the number of neighbouring nodes of  $v$ . Intuitively, high  $H(\mathcal{G})$  indicates an assortative graph and vice versa.

Table 4: Statistics of the node-classification datasets used in our experiments. The homophily level of the dataset can be used to distinguish assortative and disassortative graph datasets.

Datasets	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Cornell	Texas	Wisconsin
Homophily	0.83	0.71	0.79	0.25	0.22	0.11	0.06	0.16
Splits	140/500/1,000	120/500/1,000	60/500/1,000	60%/20%/20%	60%/20%/20%	60%/20%/20%	60%/20%/20%	60%/20%/20%
#Nodes	2,708	3,327	19,717	2,277	5,201	183	183	251
#Edges	5,429	4,732	44,338	36,101	217,073	295	309	499
#Features	1,433	3,703	500	2,325	2,089	1,703	1,703	1,703
#Classes	7	6	3	5	5	5	5	5

For the graph classification task, we adopt Mutagenicity, D&D, NCI1, Ogbg-molhiv, and QM7 datasets. The D&D and PROTEINS datasets are used for protein structure classification, which aims to categorize proteins into enzyme and non-enzyme structures. The NCI1 dataset is used for identifying chemical compounds that inhibit lung cancer cells. The Mutagenicity dataset is used for recognizing mutagenic molecular compounds that have the potential for drug development. The QM7 dataset is used for predicting the atomization energy value of molecules. The Ogbg-Molhiv is a molecular property prediction dataset for predicting whether a molecule inhibits HIV virus replication or not. All the datasets contain more than 1,000 graphs with varying graph structures (in terms of the average number of nodes and edges, the average degree of nodes) and node features. The statistics of each dataset are displayed in Table 5.

Table 5: Summary of the datasets for the graph property prediction tasks.

Datasets	PROTEINS	Mutagenicity	D&D	NCI1	ogbg-molhiv	QM7
# Graphs	1,113	4,337	1,178	4,110	41,127	7,165
Min # Nodes	4	4	30	3	2	4
Max # Nodes	620	417	5,748	111	222	23
Avg # Nodes	39	30	284	30	26	15
Avg # Edges	73	31	716	32	28	123
# Features	3	14	89	37	9	0
# Classes	2	2	2	2	2	1 (R)

### A.2 IMPLEMENTATION DETAILS

We implement our model using PyTorch. We set the default number of filters as four, which is suitable for most of the datasets. The default Chebyshev approximation order is set to 6. The dimension of hidden variables is searched from  $\{16, 32, 64\}$ , and the level of filters are selected from  $\{2, 3, 4, 5\}$ . Other hyperparameters are set at: 0.001 for the learning rate, 0.001 for weight decay, 0.5 for dropout, and 2 for the number of MM-FGConv layers. These hyper-parameters are used for both node and graph classification tasks. For the graph classification task, we further apply our proposed MM-FGPool operation as elaborated in Section 4.3, followed by a linear classifier. For baseline methods in node classification, we adopt the code from the author’s released implementation with the default settings. For graph classification, we adopt the experiment setting from Zheng et al. (2021a), where we use two-layer GCN networks followed by the pooling methods listed in Table 2. Specifically, this experiment setting is comparable to our model’s design, where we also adopt two convolutional layers and one pooling layer.

## B EXTRA EXPERIMENTS

### B.1 EXPERIMENTS ON RANDOM 60%/20%/20% SPLITS

The main results of the full sets of node classification experiments with statistics of datasets are summarized in Table 1, and Table 4. For a fair comparison with the state-of-the-art methods, we list the additional experiments for node classification on homophily graph datasets with 60%/20%/20% split. Corresponding results are shown in Table 6.

	Cora	Citeseer	Pubmed
MLP	76.44 $\pm$ 0.30	76.25 $\pm$ 0.28	86.43 $\pm$ 0.13
GCN	87.78 $\pm$ 0.96	81.39 $\pm$ 1.23	88.9 $\pm$ 0.32
GAT	76.70 $\pm$ 0.42	67.20 $\pm$ 0.46	83.28 $\pm$ 0.12
GraphSAGE	86.58 $\pm$ 0.26	78.24 $\pm$ 0.30	86.85 $\pm$ 0.11
Geom-GCN	85.27	77.99	90.05
ACM-GCN	88.62 $\pm$ 1.22	81.68 $\pm$ 0.97	90.66 $\pm$ 0.47
GCNII	88.98 $\pm$ 1.33	81.58 $\pm$ 1.3	89.8 $\pm$ 0.30
MM-FGCN (Ours)	<b>89.89*</b> $\pm$ 1.12	<b>82.97*</b> $\pm$ 0.85	<b>91.52*</b> $\pm$ 0.21

Table 6: Test accuracy for classifications on homophily graphs under 60%/20%/20% random split.

### B.2 EXPERIMENTS ON FIXED 48%/32%/20% SPLITS

We further conduct the node classification experiments on data split with fixed 48%/32%/20% according to Pei et al. (2020). The corresponding results are shown in Table 7.

	Cornell	Wisconsin	Texas	Chameleon	Squirrel	Cora	Citeseer	Pubmed
Geom-GCN	60.54 $\pm$ 3.67	64.51 $\pm$ 3.66	66.76 $\pm$ 2.72	60.00 $\pm$ 2.81	38.15 $\pm$ 0.92	85.35 $\pm$ 1.57	78.02 $\pm$ 1.15	89.95 $\pm$ 0.47
GGCN	85.68 $\pm$ 6.63	86.86 $\pm$ 3.29	84.86 $\pm$ 4.55	71.14 $\pm$ 1.84	55.17 $\pm$ 1.58	87.95 $\pm$ 1.05	77.14 $\pm$ 1.45	89.15 $\pm$ 0.37
H2GCN	82.70 $\pm$ 5.28	87.65 $\pm$ 4.98	84.86 $\pm$ 7.23	60.11 $\pm$ 2.15	36.48 $\pm$ 1.86	87.87 $\pm$ 1.20	77.11 $\pm$ 1.57	89.49 $\pm$ 0.38
MixHop	73.51 $\pm$ 6.34	75.88 $\pm$ 4.90	77.84 $\pm$ 7.73	60.50 $\pm$ 2.53	43.80 $\pm$ 1.48	87.61 $\pm$ 0.85	76.26 $\pm$ 1.33	85.31 $\pm$ 0.61
Geom-GCN	60.54 $\pm$ 3.67	64.51 $\pm$ 3.66	66.76 $\pm$ 2.72	60.00 $\pm$ 2.81	38.15 $\pm$ 0.92	85.35 $\pm$ 1.57	78.02 $\pm$ 1.15	89.95 $\pm$ 0.47
ACM-GCN	85.14 $\pm$ 6.07	88.43 $\pm$ 3.22	87.84 $\pm$ 4.4	69.14 $\pm$ 1.91	55.19 $\pm$ 1.49	87.91 $\pm$ 0.95	77.32 $\pm$ 1.7	90.00 $\pm$ 0.52
GCNII	77.86 $\pm$ 3.79	80.39 $\pm$ 3.40	77.57 $\pm$ 3.83	63.86 $\pm$ 3.04	38.47 $\pm$ 1.58	88.37 $\pm$ 1.25	77.33 $\pm$ 1.48	90.15 $\pm$ 0.43
NLMPL	84.9 $\pm$ 5.7	87.3 $\pm$ 4.3	85.4 $\pm$ 3.8	50.7 $\pm$ 2.2	33.7 $\pm$ 1.5	76.9 $\pm$ 1.8	73.4 $\pm$ 1.9	88.2 $\pm$ 0.5
NLGCN	57.6 $\pm$ 5.5	60.2 $\pm$ 5.3	65.5 $\pm$ 6.6	70.1 $\pm$ 2.9	59.0 $\pm$ 1.2	88.1 $\pm$ 1.0	75.2 $\pm$ 1.4	89.0 $\pm$ 0.5
NLGAT	54.7 $\pm$ 7.6	56.9 $\pm$ 7.3	62.6 $\pm$ 7.1	65.7 $\pm$ 1.4	56.8 $\pm$ 2.5	88.5 $\pm$ 1.8	76.2 $\pm$ 1.6	88.2 $\pm$ 0.3
MM-FGCN (Ours)	<b>87.35*</b> $\pm$ 6.18	<b>89.02*</b> $\pm$ 5.41	<b>89.31*</b> $\pm$ 1.56	<b>72.61*</b> $\pm$ 1.84	<b>61.34*</b> $\pm$ 1.22	<b>89.35*</b> $\pm$ 1.15	<b>79.86</b> $\pm$ 1.42	<b>91.42*</b> $\pm$ 0.41

Table 7: Test accuracy for node classification results under fixed 48%/32%/20% split.

### B.3 PERTURBATION RESILIENCE OF MM-FGCN.

In this study, we add extra experiments to assess the perturbation resilience of our MM-FGCN against noise perturbation present in input graph data, which is ubiquitous in real-world datasets. Particularly, we train the MM-FGCN with corrupted data that are contaminated by random noise of various magnitudes. The noise magnitude is controlled by the noise ratios, which are defined as the amount of randomly deleted edges (or randomly flipped binary-valued features) divided by the number of untainted edges (or features). We then investigate how the performance of the resultant models varies when the noise level change from 0 to 1. As illustrated in Figure 4, our MM-FGCN consistently outperforms the baselines with a remarkable margin even under the presence of considerable noise. Thus, the MM-FGCN demonstrates a strong noise resilience making it a highly promising solution for real-world applications.

### B.4 ABLATION ON THE NUMBER OF THE META-FRAMELET GENERATORS

We analyze how the size of the meta-framelet generator (i.e. a set of spectral filters),  $I$ , affects the performance of MM-FGCN. With an insufficient amount of meta-generators, the model may fail to learn the optimal frequency partition and cannot disentangle graph signals into desirable frequency components. Intuitively, an overly small  $I$  may hinder the learning of discriminative graph representations. Conversely, a large  $I$  improves the precision of frequency partition learning but also

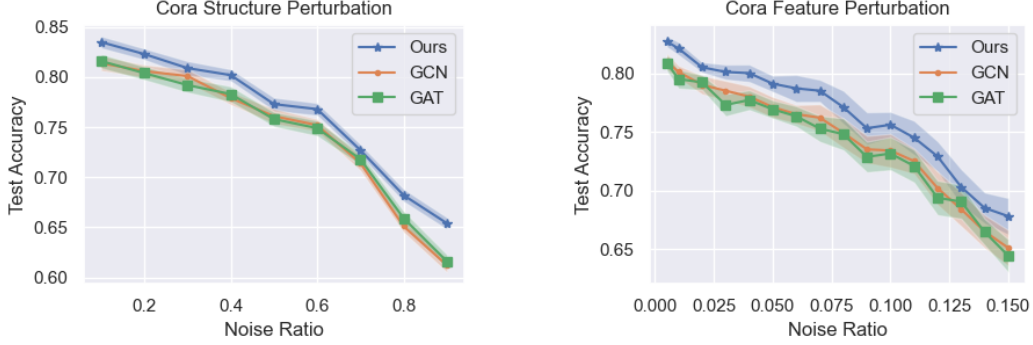


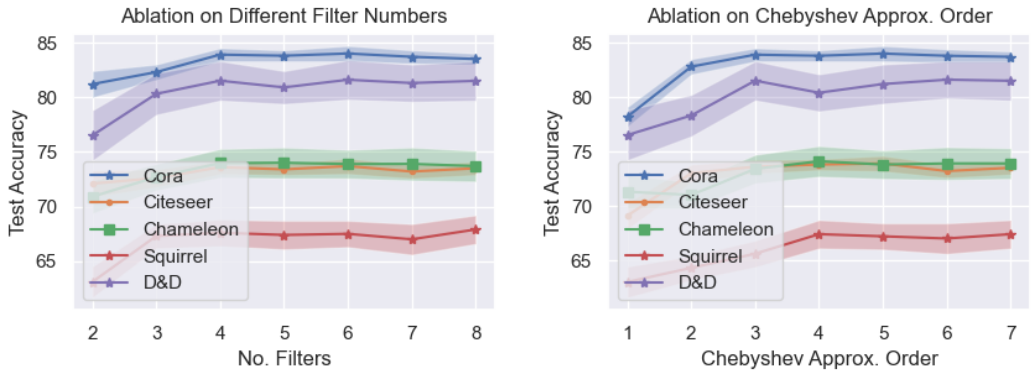
Figure 4: Noise resilience experiments with the edge (left) and feature (right) noise perturbations on Cora.

increases computational expenses. This requires us to strike a balance between the by selectively choosing the value of  $I$ .

We evaluate the performance of MM-FGCN with different choices of  $I$  over Cora, Citeseer, Chameleon, Squirrel, and D&D datasets. As shown in Figure 5 (a), the Meta-FGCN is able to achieve state-of-the-art performance by constructing the meta-framelet generator with only 3 filters. Overall, the model performance is stable and robust across different choices of  $I$ . In general, we recommend setting  $r = 4$  for effective and efficient implementation.

#### B.5 ABLATION ON THE ORDER OF CHEBYSHEV APPROXIMATION

Recall that the Chebyshev approximation trick Defferrard et al. (2016b) is applied in Algorithm 1 for efficient computation of each  $g_{r,i}^\omega(\mathbf{L})$ . Broadly speaking, using a higher-order Chebyshev approximation leads to a smaller approximation error w.r.t the meta-generator, but creates a greater computational overhead. As illustrated in Figure 5 (b), the performance of MM-FGCN is robust to Chebyshev approximation of different orders. Empirically, the MM-FGCN achieves optimal performance with a Chebyshev approximation of an order greater than 4. In contrast, a low-order Chebyshev approximation order incurs an undesirable approximation error, which hinders graph representation learning and impairs model generalization. We recommend using a higher than 4-order Chebyshev approximation for good model performance.



(a) Ablation on the number of filters used in the meta-framelet generator.

(b) Ablation on the order of Chebyshev approximation.

Figure 5: Ablation studies on MM-FGCN’s hyperparameters.

#### B.6 VISUALIZATION OF MM-FGCN REPRESENTATION

In this section, we empirically show that the MM-FGCN is able to produce more discriminative graph representations than conventional GCN, on both assortative (e.g. Cora) and disassortative (e.g.

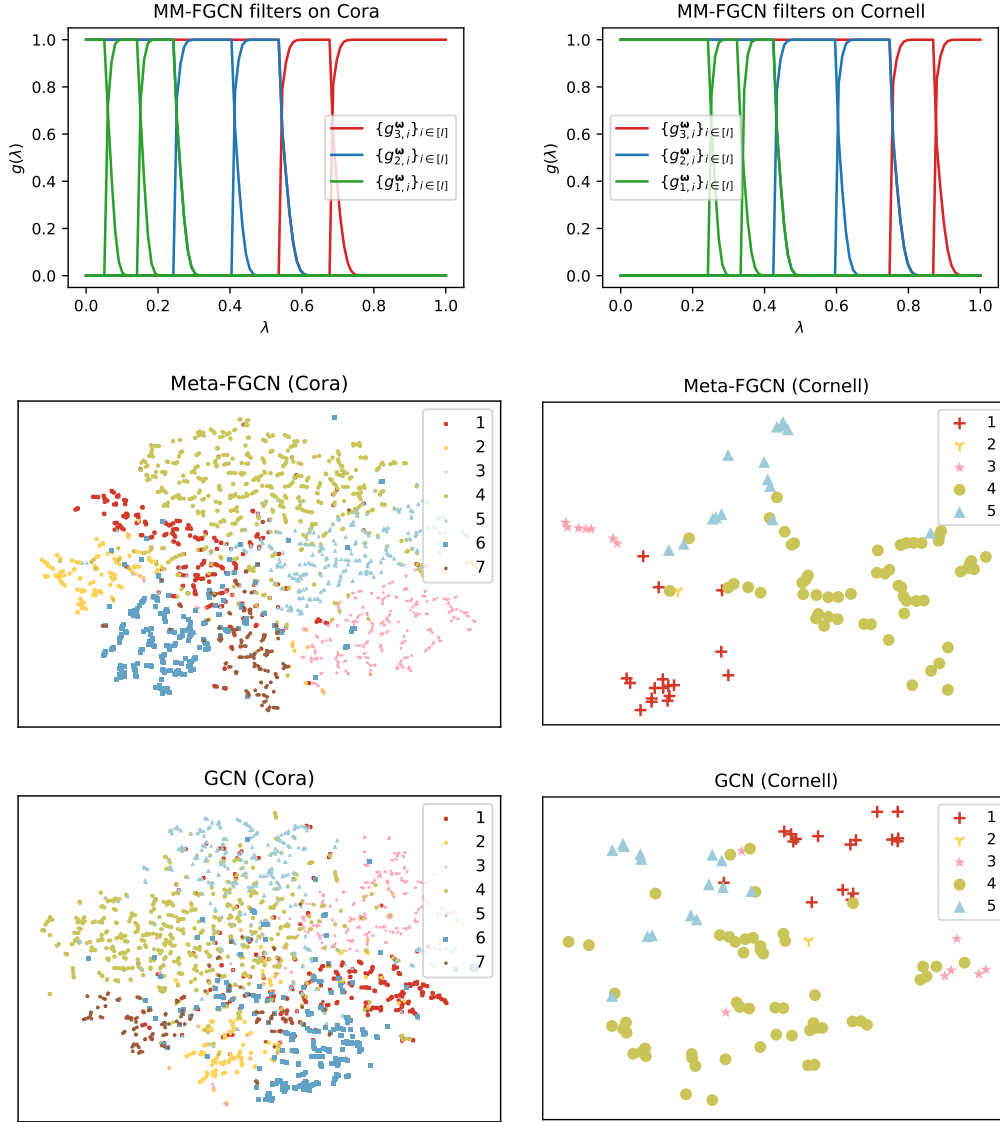


Figure 6: Meta-framelet generator (row 1) and feature visualization (row 2-3) on the test and validation sets of Cora (left, assortative) and Cornell (right, disassortative) datasets using MM-FGCN and GCN.

Cornell) datasets. In order to assess the quality of the learned graph representation, we visualize the hidden features generated by the penultimate layer of both MM-FGCN and GCN via t-SNE van der Maaten & Hinton (2008). As shown in Figure 6, the graph representation of our MM-FGCN is more spatially clustering than GCN, especially for the disassortative dataset that is more challenging for classification. In fact, the learned feature of MM-FGCN is strongly correlated to the label, which significantly facilitates node and graph classification tasks. In summary, the visualization of the learned hidden features validates that the adaptiveness and expressiveness of our MM-FGCN are beneficial to learning discriminative graph representations.

We also demonstrate the effectiveness of the MM-FGCN by visualising the filters learned by MM-FGCN in Figure 6. We can observe that the filters learned from the Cora dataset (assortative) more concentrate on the low-frequency signals than the filters learned from the Cornell dataset (disassortative). Due to the assortative properties, aggregating local information with low-frequency signals can benefit the model’s performance on the Cora dataset. In contrast, as elaborated in Bo et al. (2021), high-frequency signals are useful for disassortative networks, which corresponds to our learned multi-resolution filters since more filters concentrate on the high-frequency part, providing a comprehensive feature extraction on the high-frequency signal. This phenomenon shows the adaptivity of our MM-FGCN on different types of graphs.

## B.7 ROBUSTNESS ANALYSIS

We verify that our MM-FGCN inherently possesses greater robustness compared to conventional GCN models, even without relying on any specific robust data augmentation techniques. Furthermore, we observe that the robustness of our MM-FGCN can be further enhanced when combined with data augmentation methods. We assess the robustness of our MM-FGCN in the face of both adaptive and non-adaptive graph attacks Mujkanovic et al. (2022). These attacks encompass graph poisoning, which involves perturbing the adjacency matrix before training, and graph evasion, which perturbs the adjacency matrix after training. We not only compare our method with GCN but also evaluate two prevalent graph defense techniques, namely GRAND defense and Jaccard defense. We present the results on Cora in the top row of Figure 7, demonstrating that our MM-FGCN showcases substantially higher adversarial accuracy than GCN when subjected to graph attacks. Moreover, by incorporating GRAND and Jaccard defense techniques, the adversarial accuracy of MM-FGCN can be further enhanced. This highlights that the inherent robustness gain of MM-FGCN is independent of the benefits offered by graph defense methods, demonstrating our orthogonal contributions.

We also validate our MM-FGCN’s immunity against realist graph topology perturbations. To evaluate the robustness of a well-trained model, we introduce noise of varying magnitudes into the adjacency matrix. Subsequently, we assess the deviation between the perturbed hidden representation and the original one using the  $\ell_2$  distance metric. As depicted in the bottom row of Figure 7, our MM-FGCN demonstrates a distinct ‘potential well’, where the potential refers to the distance from the perturbed feature to the original, unperturbed feature. The results in Figure 7 show that our MM-FGCN has the capability to learn robust representations that remain intact by minor perturbations in the graph adjacency matrix. In contrast, conventional GCN experiences a progressively linear increase in representation distortion as the magnitude of perturbation grows.

## C PROOF DETAILS

*Proof of Proposition 2.* According to the discrete tight framelet transform theory (Theorem 2.1 and Theorem 3.1 in (Dong, 2017a)), the series of progressive resolution subspaces  $\{V_r\}_r$  with  $V_r = \text{span}(\{\varphi_{riv}\}_{i,v})$  inherently satisfies the denseness, translation, and dilation properties, making it a set of desirable multiresolution bases for graph domain data. To complete the proof, one only need to verify the tightness of the MMFS, i.e.  $\Phi_{\text{MM}}\Phi_{\text{MM}}^\top \mathbf{x} = \mathbf{x}$  holds for any graph signal  $\mathbf{x}$ . By

definition, we have

$$\Phi_{MM} \Phi_{MM}^\top = \sum_{r,i,v} \varphi_{riv} \varphi_{riv} \quad (10)$$

$$= \sum_{r,i} (\mathbf{U} g_{r,i}^\omega (\gamma^{-J+r} \mathbf{L}) \mathbf{U}^\top) (\mathbf{U} g_{r,i}^\omega (\gamma^{-J+r} \mathbf{L}) \mathbf{U}^\top)^\top \quad (11)$$

$$= \mathbf{U} \left( \sum_{r,i} g_{r,i}^{\omega^2} (\gamma^{-J+r} \mathbf{L}) \right) \mathbf{U}^\top \quad (12)$$

$$= \mathbf{U} \left( \sum_{r \in [R]} \sum_{i \in [I]} (g_{1,i}^{\omega^2} (\gamma^{-J+r} \mathbf{L})) g_{1,1}^{\omega^2} (\gamma^{-J+r-1} \mathbf{L}) \dots g_{1,1}^{\omega^2} (\gamma^{-J+r} \mathbf{L}) \right) \mathbf{U}^\top \quad (13)$$

$$= \mathbf{U} \left( \sum_{r \in [R]} g_{1,1}^{\omega^2} (\gamma^{-J+r-1} \mathbf{L}) \dots g_{1,1}^{\omega^2} (\gamma^{-J+r} \mathbf{L}) \right) \mathbf{U}^\top \quad (14)$$

$$= \mathbf{U} \left( \sum_{r \in [R]} g_{1,1}^{\omega^2} (\gamma^{-J+r-1} \mathbf{L}) \dots g_{1,1}^{\omega^2} (\gamma^{-J+r} \mathbf{L}) \right) \mathbf{U}^\top \quad (15)$$

■

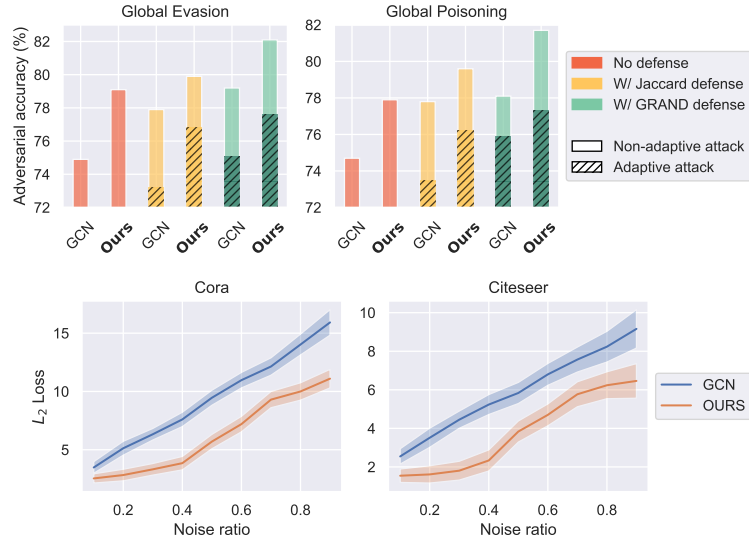


Figure 7: Top: the adversarial accuracy of MM-FCGN under graph attack. Bottom: representation distortion of each perturbation level.