


IGL-BENCH: ESTABLISHING THE COMPREHENSIVE BENCHMARK FOR IMBALANCED GRAPH LEARNING

Jiawen Qin[1*], Haonan Yuan[1*], Qingyun Sun[1*], Lyujin Xu[1], Jiaqi Yuan[1], Pengfeng Huang[1], Zhaonan Wang[1], Xingcheng Fu[2], Hao Peng[1], Jianxin Li[1†], Philip S. Yu[3]

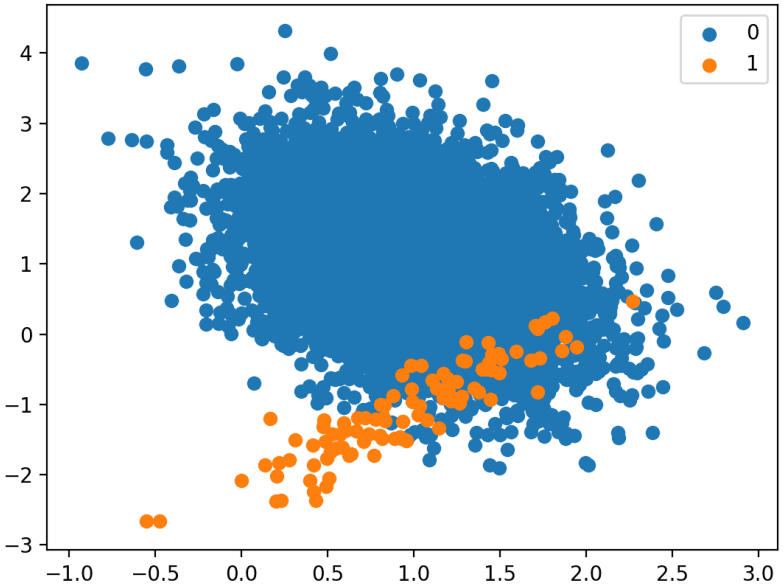
[1]Beihang University

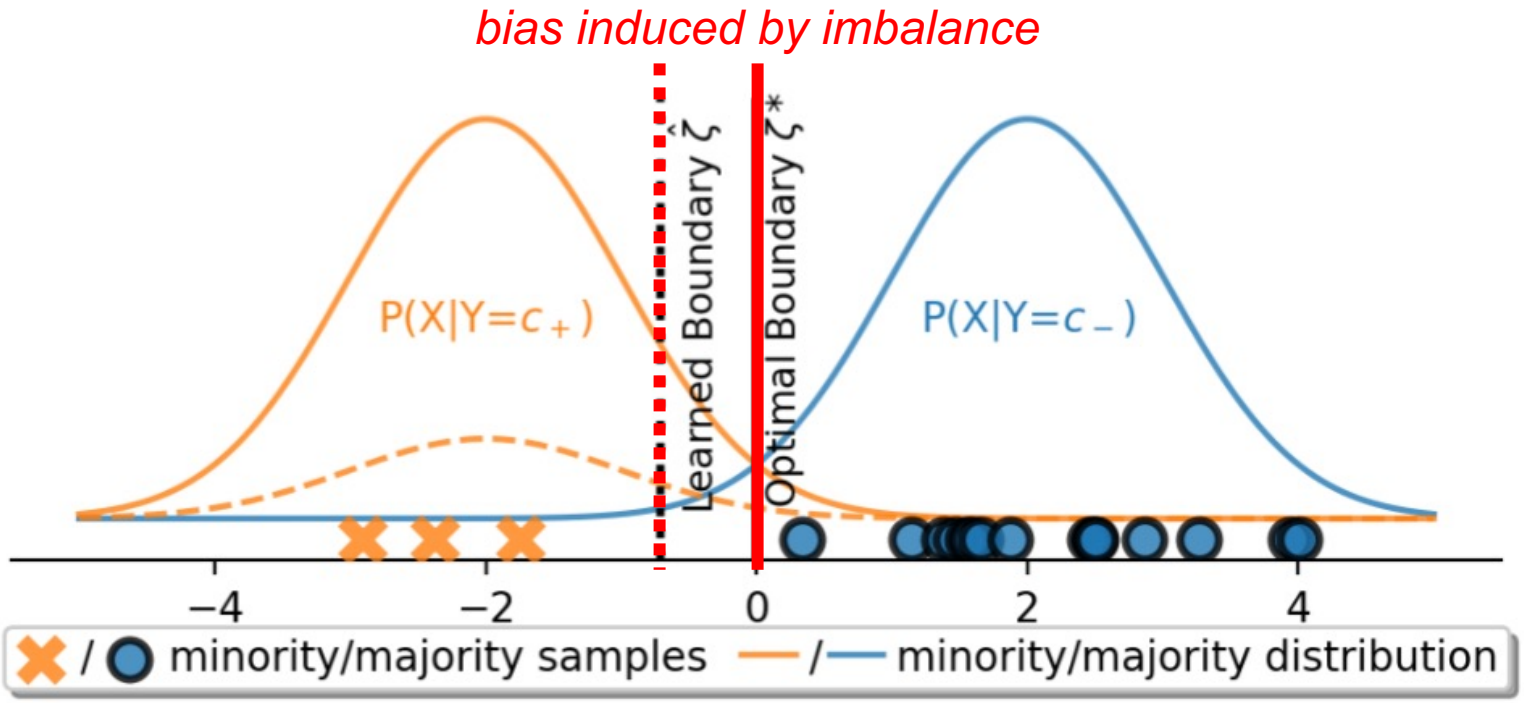
[2]Guangxi Normal University

[3]University of Illinois Chicago

Imbalance Problem in Machine Leaning

- Data imbalance leads to decision boundary shift



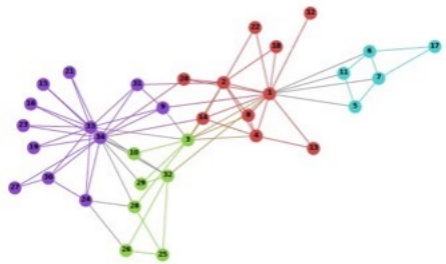


[1] https://arxiv.org/pdf/2111.12791.pdf

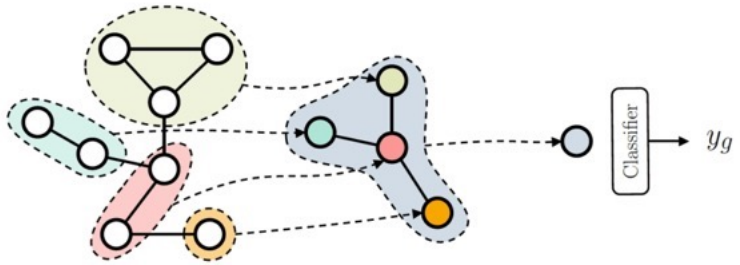
Graphs: Core of Real-World Data Structures

- Graph Neural Networks (GNNs) have been proposed to tackle highly complex structural information

social network



node-level representation learning



graph-level representation learning

node classifcation

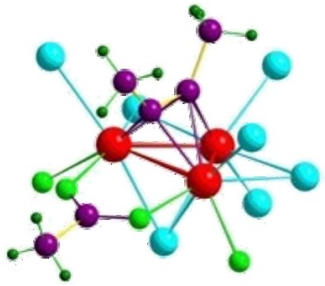
link prediciton

.....

graph classifcation

graph regression

.....



molecular network

Imbalance Issues in Graphs

- Class Imbalance and Topology Imbalance (Local/Global)
- Node Level and Graph Level

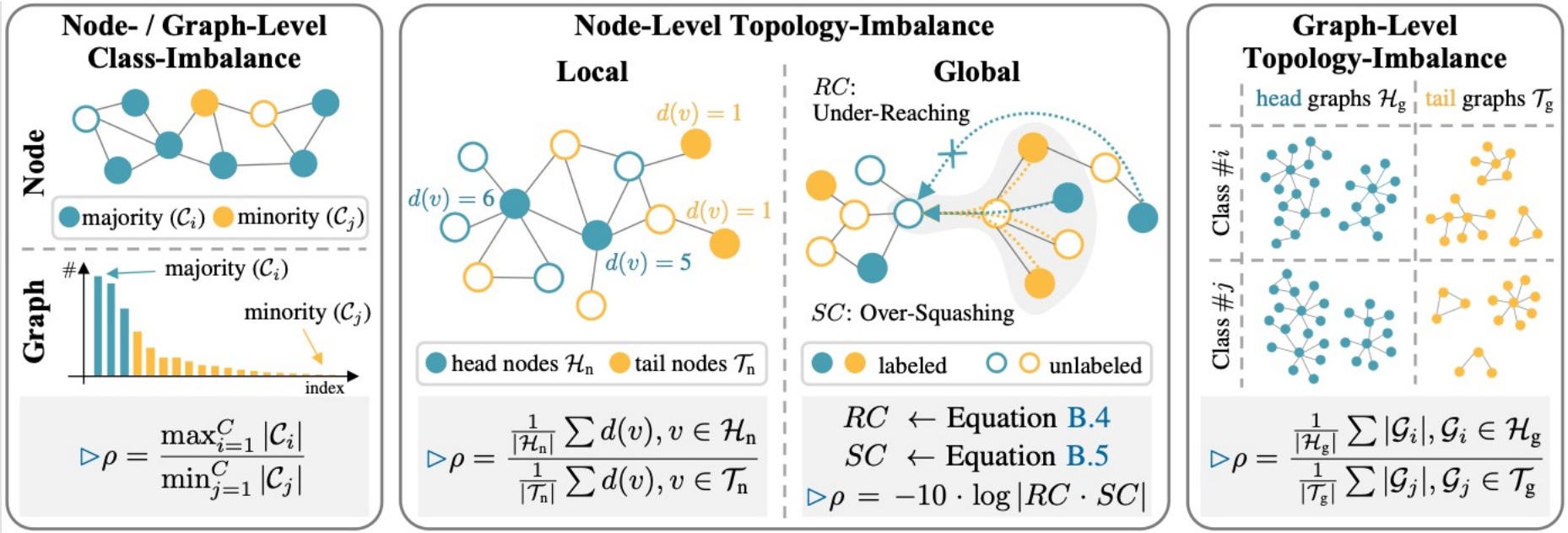
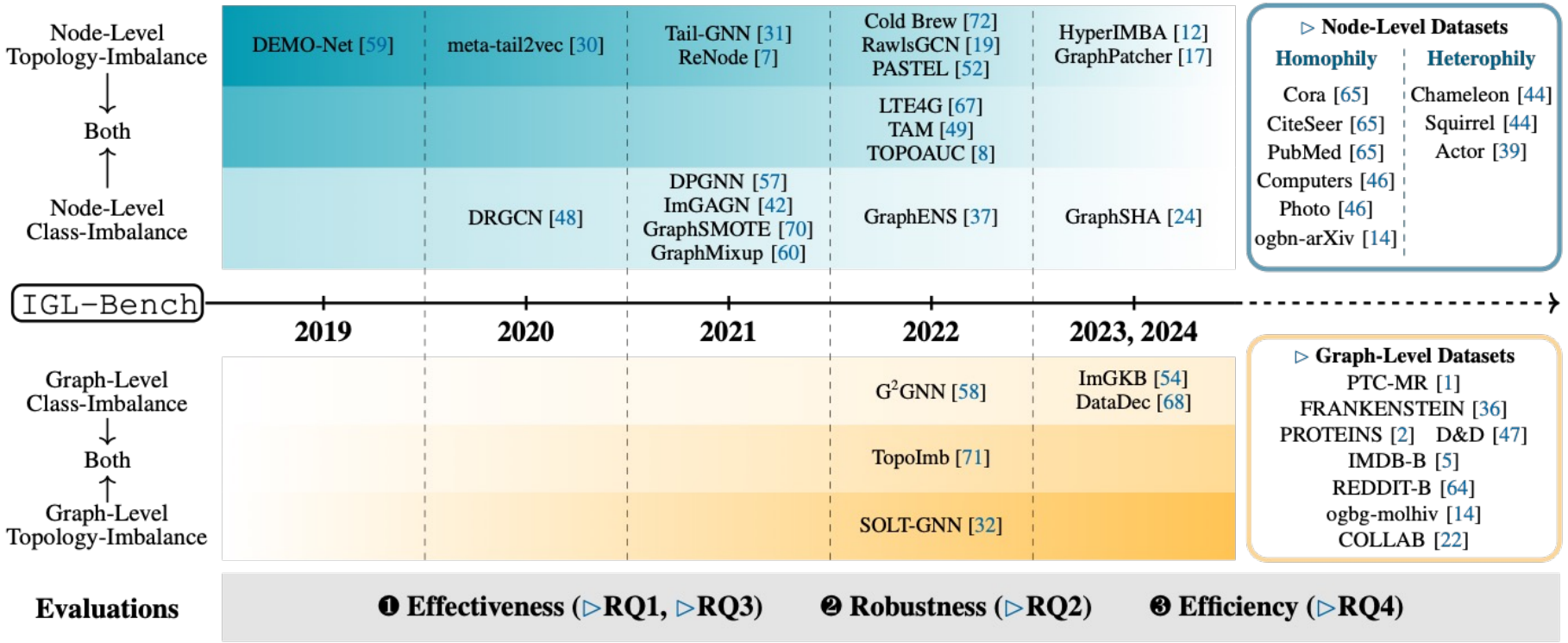


Figure 2: The research scope of the proposed `IGL-Bench`. Definitions of the imbalance ratio (ρ) corresponding to each imbalance issue are further concluded in Table 1. Click $\triangleright$ and check details.

Overview of IGL-Bench

- IGL-Bench embarks on 17 diverse graph datasets and 24 distinct IGL algorithms.
- IGL-Bench investigates IGL algorithms in terms of effectiveness, robustness, and efficiency on node-level and graph-level tasks.



Imbalanced Datasets Settings in IGL-Bench

- IGL-Bench defines imbalance ratios (High/Mid/Low) for different imbalance types.
- IGL-Bench emarks on uniform data processing and splitting strategies.

Table 1: Definitions of the imbalance ratio (ρ) across different imbalance types.

Imbalance Type	Definition	Explanation
Node-Level Class-Imbalance Graph-Level Class-Imbalance	$\rho = \frac{\max_{i=1}^{C} \lvert\mathcal{C}_i\rvert}{\min_{j=1}^{C} \lvert\mathcal{C}_j\rvert}$	The imbalance ratio is set to the ratio between the number of samples ($\lvert\mathcal{C}\rvert$) in the majority and the minority class.
Node-Level Topology-Imbalance (local and global)	$\rho = \frac{\frac{1}{\lvert\mathcal{H}_n\rvert}\sum d(v), v \in \mathcal{H}_n}{\frac{1}{\lvert\mathcal{T}_n\rvert}\sum d(v), v \in \mathcal{T}_n}$	The **local** imbalance ratio is set to the ratio of the average node degree ($d(v)$) of the head node set ($\mathcal{H}_n$) to the average node degree of the tail node set ($\mathcal{T}_n$).
	$\rho = -10 \cdot \log \lvert RC \cdot SC \rvert$	The **global** imbalance ratio is set to the negative logarithm of the absolute value of the product of the Reaching Coefficient (RC) and the Squashing Coefficient (SC).
Graph-Level Topology-Imbalance	$\rho = \frac{\frac{1}{\lvert\mathcal{H}_g\rvert}\sum \lvert\mathcal{G}_i\rvert, \mathcal{G}_i \in \mathcal{H}_g}{\frac{1}{\lvert\mathcal{T}_g\rvert}\sum \lvert\mathcal{G}_j\rvert, \mathcal{G}_j \in \mathcal{T}_g}$	The imbalance ratio is set to the ratio of the average graph size (number of nodes) of the head graph set ($\mathcal{H}_g$) to the average graph size of the tail graph set ($\mathcal{T}_g$).

RQ1: How much progress has been made by the existing IGL algorithms?

- **Key Insights 1:** Node-level class-imbalance and topology-imbalance often coexist.
- **Key Insights 2:** For node-level classification, the homophily or heterophily property of the significantly impacts the learning on class-imbalanced and topology-imbalanced graphs. There is a lack of effective algorithms that address both types of imbalance in large-scale graphs without relying on homophily assumptions.

Table 2: **Accuracy** score (% ± standard deviation) of **node** classification on manipulated **class-imbalanced** graph datasets (**Low**) over 10 runs. "—" denotes out of memory or time limit. The best results are shown in **bold** and the runner-ups are underlined (the same for tables below).

Algorithm	Cora 0.81	CiteSeer 0.74	PubMed 0.80	Computers 0.78	Photo 0.82	ogbn-arXiv 0.65	Chameleon 0.23	Squirrel 0.22	Actor 0.22
GCN (bb.) [21]	76.36±0.13	52.96±0.55	60.57±0.19	75.06±0.50	69.80±6.15	59.83±0.23	26.35±0.24	17.16±0.17	24.06±0.14
DRGCN [48]	71.35±0.77	55.22±1.82	62.59±4.62	67.71±3.10	85.67±5.30	—	26.40±0.35	17.11±0.81	25.03±0.23
DPGNN [57]	72.91±3.95	56.78±2.23	81.87±2.80	68.69±8.62	81.66±9.19	—	30.58±1.48	**25.35±1.48**	21.66±1.68
ImGAGN [42]	73.48±3.07	55.29±3.00	72.16±1.51	74.92±1.87	83.10±3.42	—	24.38±2.86	18.75±1.80	24.54±3.38
GraphSMOTE [70]	77.21±0.27	53.55±0.95	71.25±0.27	70.54±1.52	89.07±1.12	—	27.23±0.21	16.79±0.14	25.08±0.31
GraphENS [37]	79.34±0.49	61.98±0.76	80.84±0.17	80.72±0.68	90.38±0.37	53.23±0.52	24.34±1.62	20.05±1.61	25.03±0.38
GraphMixup [60]	79.88±0.43	62.66±0.70	75.94±0.09	**86.15±0.47**	89.69±0.31	56.08±0.31	30.95±0.40	17.83±0.32	24.75±0.37
LTE4G [67]	80.53±0.65	64.48±1.56	**83.02±0.33**	79.35±1.39	87.94±1.82	—	31.91±0.34	19.37±0.41	**25.43±0.26**
TAM [49]	80.69±0.27	64.16±0.24	81.47±0.15	81.30±0.53	90.35±0.42	53.49±0.54	23.27±1.38	21.17±0.95	24.53±0.33
TOPOAUC [8]	**83.34±0.31**	**69.03±1.33**	—	70.85±4.55	83.72±2.23	—	**33.60±1.51**	21.38±1.03	25.16±0.46
GraphSHA [24]	80.03±0.46	60.51±0.61	77.94±0.36	82.71±0.40	**91.55±0.32**	**60.30±0.13**	23.73±1.97	20.05±1.61	23.59±1.01

Table 3: **Accuracy** score (% ± standard deviation) of **node** classification on manipulated **local topology-imbalanced** graph datasets (**Mid**) over 10 runs. "—" denotes out of memory or time limit.

Algorithm	Cora 0.81	CiteSeer 0.74	PubMed 0.80	Computers 0.78	Photo 0.82	ogbn-arXiv 0.65	Chameleon 0.23	Squirrel 0.22	Actor 0.22
GCN (bb.) [21]	80.16±1.09	66.87±0.85	83.97±0.13	71.65±2.10	89.43±0.58	52.93±0.33	52.74±0.60	28.70±0.68	21.55±1.74
DEMO-Net [59]	80.37±0.52	69.73±1.31	84.11±0.20	79.38±0.98	88.09±1.30	65.81±0.11	55.51±0.87	39.45±0.62	29.12±0.30
meta-tail2vec [30]	32.17±0.68	29.97±3.61	59.82±2.86	68.17±1.07	79.82±1.02	33.71±1.16	38.78±0.44	24.90±0.25	26.09±0.07
Tail-GNN [31]	79.05±1.15	69.97±1.03	85.78±0.41	**84.09±1.01**	92.21±0.09	—	53.20±0.80	30.43±1.06	28.02±0.71
Cold Brew [72]	73.84±2.10	67.42±0.97	**86.51±0.04**	80.19±0.24	88.13±0.24	**69.97±0.07**	**59.16±0.40**	**43.04±0.24**	**33.01±0.19**
LTE4G [67]	82.54±0.46	70.55±0.54	84.77±0.78	81.32±2.21	91.09±0.19	—	55.84±2.86	32.43±3.31	24.00±0.49
RawlsGCN [19]	80.52±0.14	72.38±0.43	86.05±0.12	78.78±1.40	90.53±1.32	40.00±0.05	44.96±0.79	29.93±0.65	28.29±0.24
GraphPatcher [17]	**83.25±0.42**	**73.38±0.42**	85.60±0.16	83.68±0.69	**92.28±0.06**	66.74±0.04	55.19±0.41	36.94±0.11	23.85±0.92

Table 4: **Accuracy** score (% ± standard deviation) of **node** classification on manipulated **global topology-imbalanced** graph datasets (**High**) over 10 runs.

Algorithm	Cora 0.81	CiteSeer 0.74	PubMed 0.80	Computers 0.78	Photo 0.82	ogbn-arXiv 0.65	Chameleon 0.23	Squirrel 0.22	Actor 0.22
GCN (bb.) [21]	79.10±1.28	68.37±1.73	83.44±0.16	75.02±2.20	86.32±1.90	51.04±0.18	33.90±0.70	23.27±0.82	22.40±0.68
ReNode [7]	79.91±1.52	69.89±0.73	82.97±0.12	77.95±1.71	87.80±0.52	50.68±0.15	32.92±0.98	23.80±0.59	22.39±0.62
TAM [49]	80.50±0.18	**73.14±0.13**	84.07±0.12	82.35±0.19	89.80±0.23	**52.09±0.06**	35.64±0.27	24.58±0.09	22.55±0.06
PASTEL [52]	**80.91±0.36**	72.73±0.26	—	83.24±0.85	89.10±0.41	—	**47.12±2.82**	**33.15±0.66**	**27.56±1.04**
TOPOAUC [8]	79.27±0.52	70.08±0.83	—	75.35±1.32	87.10±0.98	—	33.39±2.09	22.86±0.36	22.56±0.18
HyperIMBA [12]	79.81±0.78	71.78±0.40	**84.75±0.30**	**83.43±0.65**	**90.65±0.14**	—	38.30±2.70	29.97±1.79	25.30±2.56

Table 5: **Accuracy** score (% ± standard deviation) of **graph** classification on manipulated **class-imbalanced** graph datasets (**Low**) over 10 runs.

Algorithm	PTC-MR	FRANKENSTEIN	PROTEINS	D&D	IMDB-B	REDDIT-B	ogbg-molhiv	COLLAB
GIN (bb.) [63]	47.83±2.95	63.38±1.93	55.38±3.57	51.05±5.07	62.31±3.99	61.10±4.86	60.75±3.79	65.01±1.33
G²GNN [58]	51.88±6.23	61.13±1.05	63.61±5.03	56.29±7.30	63.87±4.64	69.58±3.59	**65.00±3.81**	62.05±3.06
TopoImb [71]	44.86±3.52	49.49±7.14	52.12±10.51	49.97±7.24	59.95±5.19	59.67±7.30	—	65.88±0.75
DataDec [68]	**55.72±2.88**	**67.99±0.75**	66.58±1.35	63.51±1.62	**67.92±3.37**	**78.39±5.01**	—	**71.48±1.03**
ImGKB [54]	50.11±5.95	40.83±0.02	**66.60±2.64**	**65.85±3.70**	47.74±0.29	67.50±2.70	48.57±2.14	51.21±0.10

Table 6: **Accuracy** score (% ± standard deviation) of **graph** classification on manipulated **topology-imbalanced** graph datasets (**Mid**) over 10 runs.

Algorithm	PTC-MR	FRANKENSTEIN	PROTEINS	D&D	IMDB-B	REDDIT-B	ogbg-molhiv	COLLAB
GIN (bb.) [63]	51.38±6.78	54.82±2.26	62.14±2.43	61.46±2.43	65.08±5.78	68.32±1.77	57.67±3.12	65.84±3.12
SOLT-GNN [32]	**53.04±3.91**	**68.71±1.60**	**71.95±2.36**	63.33±1.86	**69.38±1.23**	**73.51±1.14**	—	**69.69±2.45**
TopoImb [71]	51.59±4.30	54.52±0.87	64.03±4.43	**65.99±1.25**	68.10±0.87	71.54±0.75	—	68.68±1.34

- ***RQ2: How effective are the IGL algorithms generalizing to the changing imbalance ratio?***
  - **Key Insights 1:** As the imbalance degree increases, the performance tends to degrade, especially under extreme conditions.
  - **Key Insights 2:** Class-imbalance and topology-imbalance do not seem to be entirely orthogonal issues. Future research should further investigate the impact of topology and class imbalance on each other in imbalanced graph learning by analyzing their intrinsic causes.

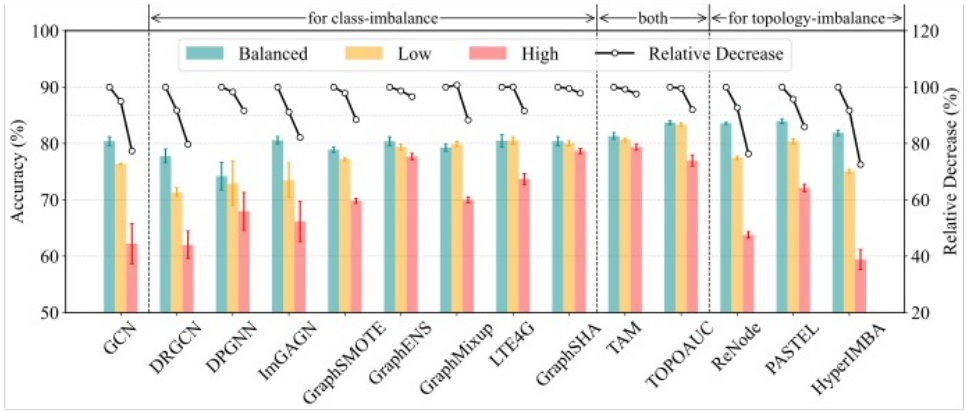


Figure 3: Robustness analysis of **node-level** algorithms under different **class-imbalance** levels on **Cora** (homophilic). Results are **Accuracy** and its relative decrease compared to the balanced split.

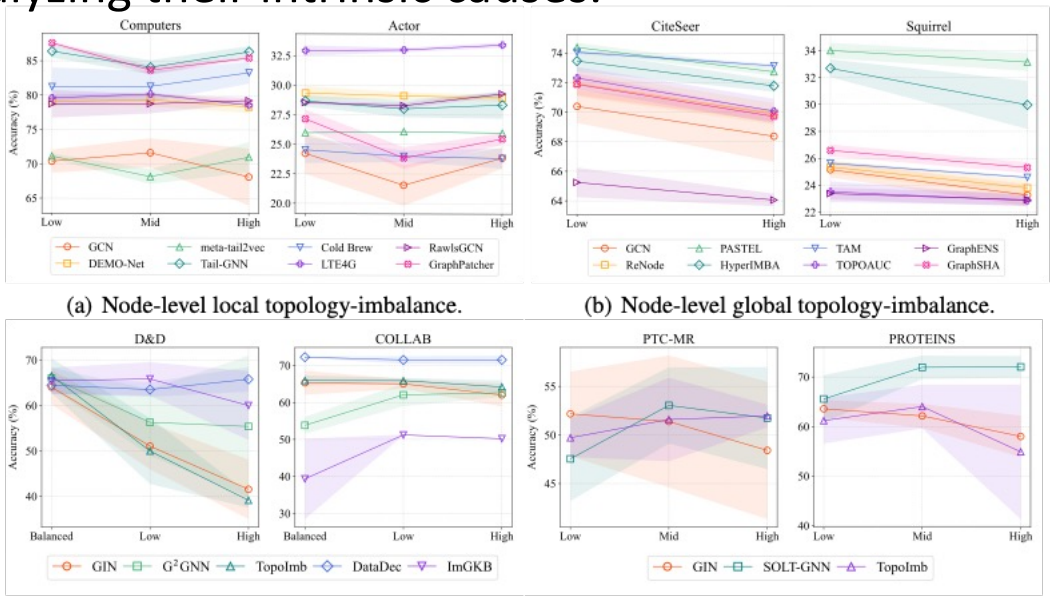
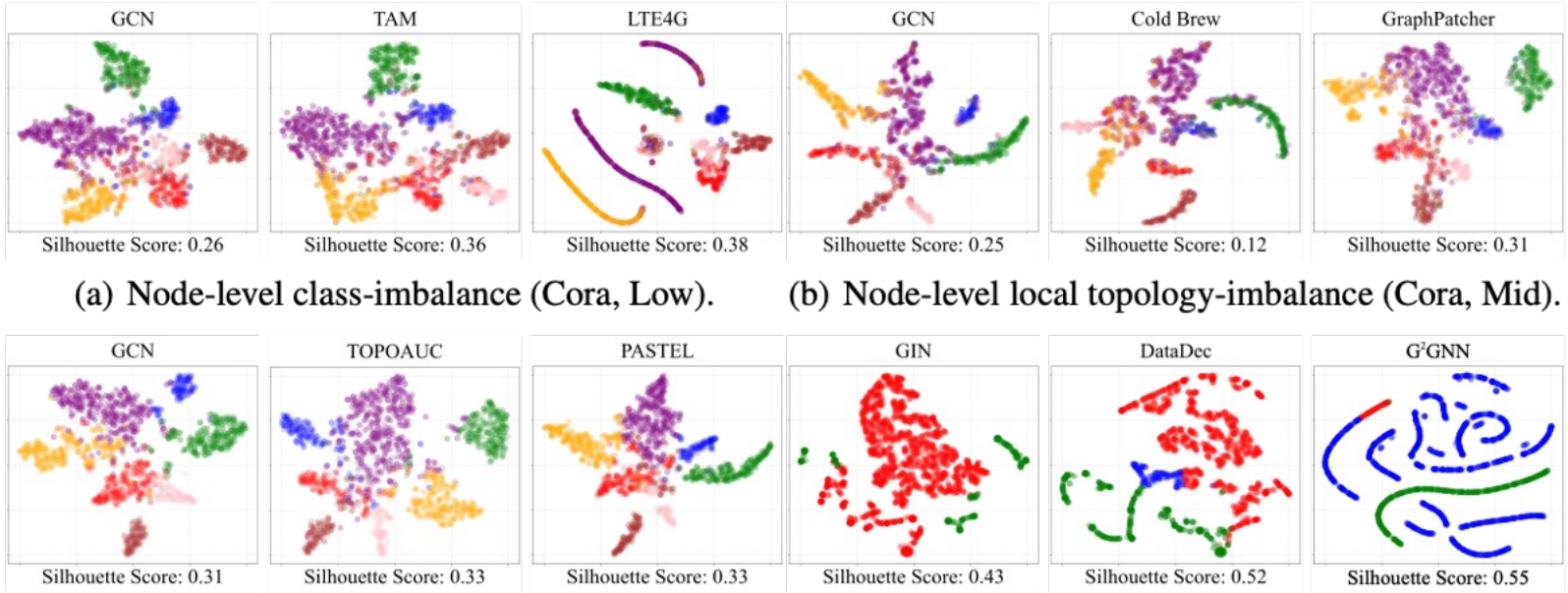


Figure 4: Robustness analysis of the **node-level** and **graph-level** algorithms under different imbalance levels. Results are reported with the algorithm performance (**Accuracy**) with the standard deviation.

- ***RQ3: Does classifiers benefit from the IGL algorithms to learn clearer boundaries?***
  - **Key Insights:** Future research should focus more on exploring dynamic methods to adjust boundary sensitivity in response to imbalanced data, which could further enhance classification performance.



(a) Node-level class-imbalance (Cora, Low). (b) Node-level local topology-imbalance (Cora, Mid).

(c) Node-level global topology-imbalance (Cora, High). (d) Graph-level class-imbalance (COLLAB, Low).

Figure 5: Visualization of node- and graph-level IGL algorithms in varying imbalanced scenarios.

RQ4: How efficient are these IGL algorithms in terms of time and space?

- **Key Insights:** Graph learning for ultra-large-scale data is a prominent research frontier in the community. The new paradigm of IGL models poses substantial challenges in memory-time-efficiently addressing imbalanced graph data and achieving high-quality representation learning.

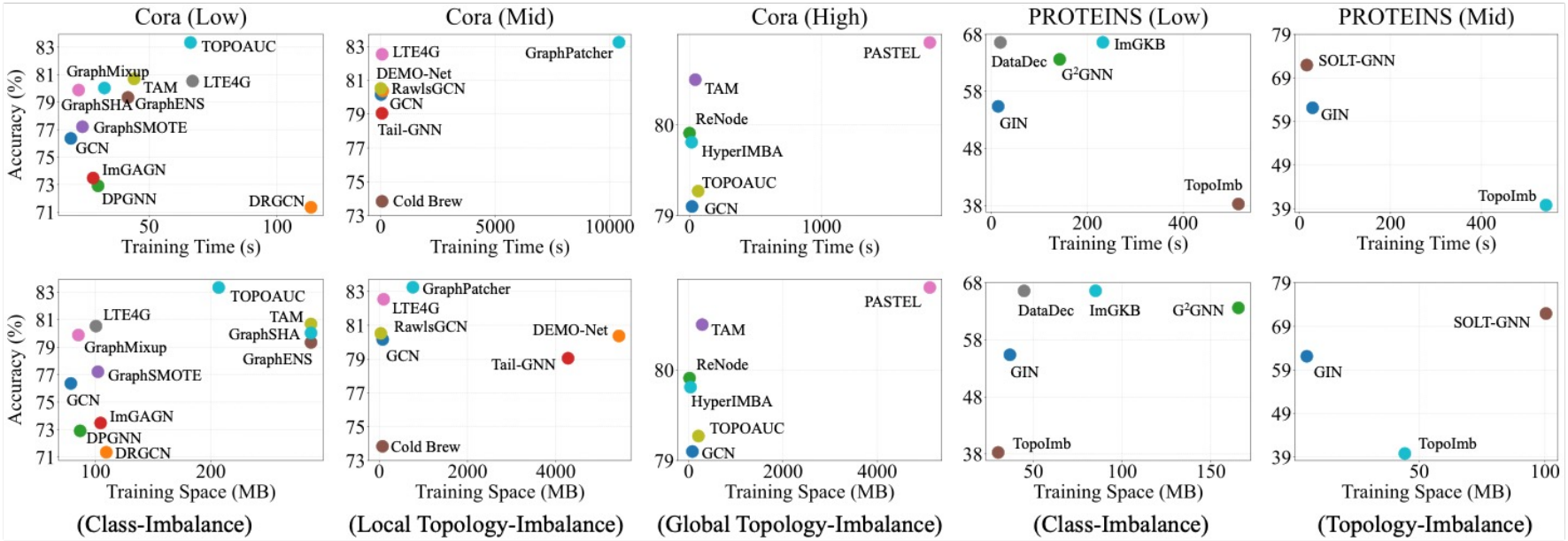


Figure 6: Time and space analysis of node- and graph-level IGL algorithms on Cora and Proteins.

Future Directions

- **Unified Algorithm**. Class-imbalance and topology-imbalance simultaneously and widely exist in multi-domain graphs. Future research should revisit the optimization conflicts between two imbalance issues and develop a unified “one for both” IGL algorithm rooted in core nature of the problem.
- **Robustness and Generalization**. Future research should emphasize enhancing the robustness of IGL algorithms in extreme imbalance scenarios and improving their generalization to handle unseen testing domains or unprecedented distribution shifts, ensuring reliable performance in diverse real-world settings.
- **Efficiency and Scalability.** As the size of graphs continues to grow exponentially, a key area of future research is the reduction of memory and computational complexity in IGL algorithms to ensure their efficient scalability and performance on large-scale graphs.