Position-aware Structure Learning for Graph Topology-imbalance by Relieving Under-reaching and Over-squashing

Qingyun Sun
Beihang University
Beijing, China
sunqy@buaa.edu.cn

Jianxin Li
Beihang University
Beijing, China
lijx@buaa.edu.cn

Haonan Yuan
Beihang University
Beijing, China
yuanhn@act.buaa.edu.cn

Xingcheng Fu
Beihang University
Beijing, China
fuxc@act.buaa.edu.cn

Hao Peng
Beihang University
Beijing, China
penghao@act.buaa.edu.cn

Cheng Ji
Beihang University
Beijing, China
jicheng@act.buaa.edu.cn

Qian Li
Beihang University
Beijing, China
liqian@act.buaa.edu.cn

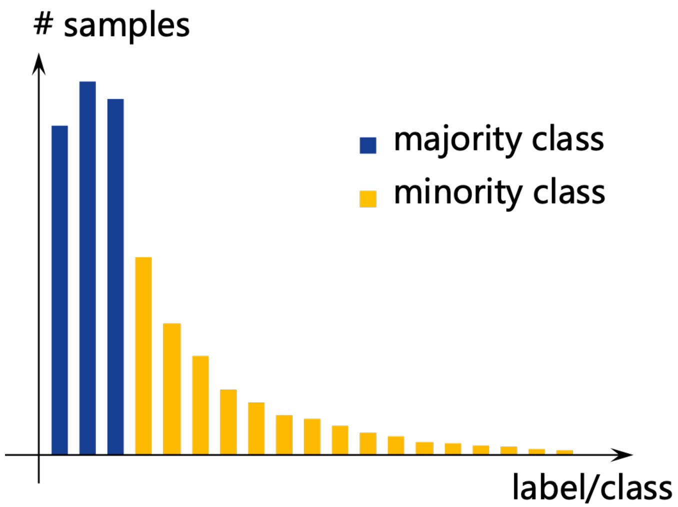
Philip S. Yu
University of Illinois Chicago
Chicago, USA
psyu@uic.edu

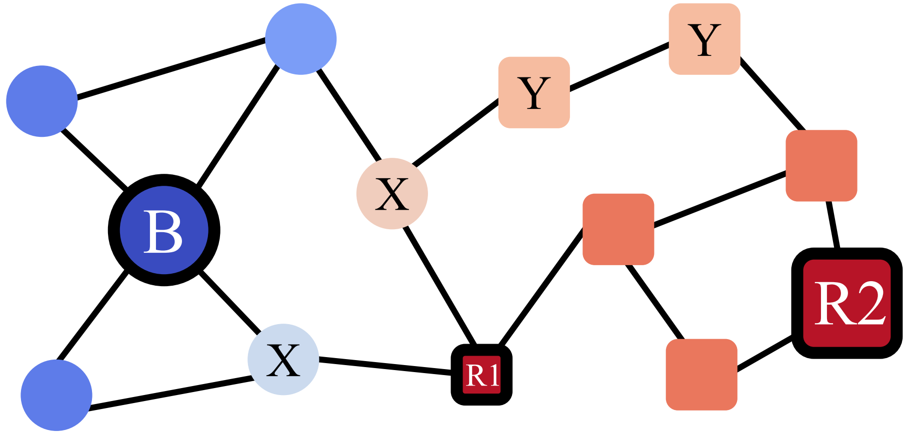
paper: https://arxiv.org/pdf/2208.08302.pdf
code: https://github.com/RingBDStack/PASTEL

2022.10.18

Imbalance Learning

- **Class-imbalance (Quantity-imbalance)**
  - performance is dominated by the **majority class**
  - **Re-sampling:** balance the number *(data-level)*
  - **Re-weighting:** adjust sample weights *(algorithm-level)*
- **Topology-imbalance (graph-specific, node classification)**
  - asymmetric & uneven topology positions of **labeled nodes**





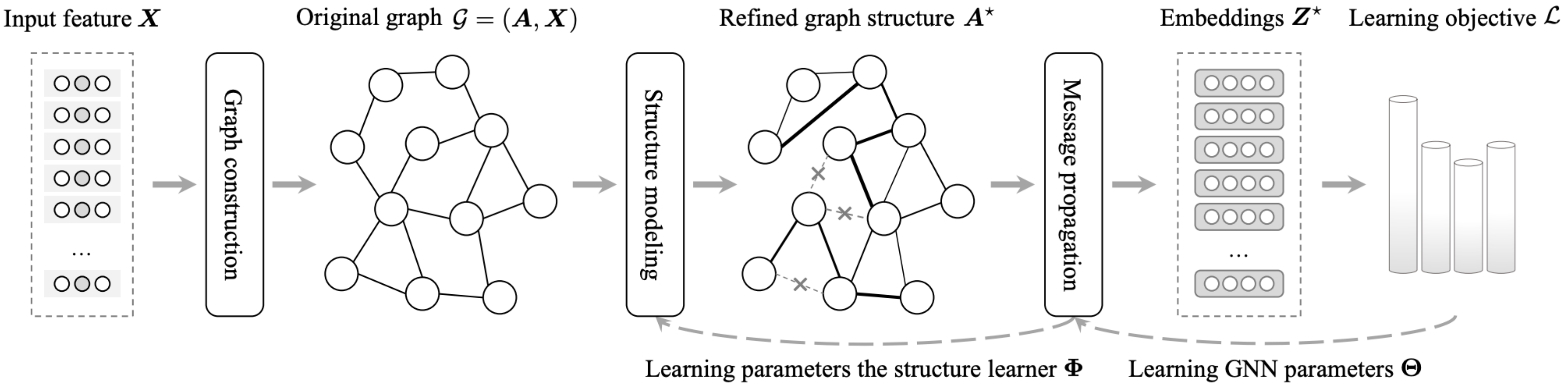
- **ReNode**
  - classical **re-weighting on node loss**
  - directed by an influence conflict detection based metric
- Other method? **Graph Structure Learning!**

Acknowledgement from https://arxiv.org/abs/2110.04099

Graph Structure Learning (GSL)

- **Target:** jointly learning an optimized **structure** and **corresponding representations** to improving the robustness of GNN models.
- **Paradigm: Structure Modeling → Message Passing → Learning Objective**

Input: raw graph **Output:** refined graph (topology-balanced)

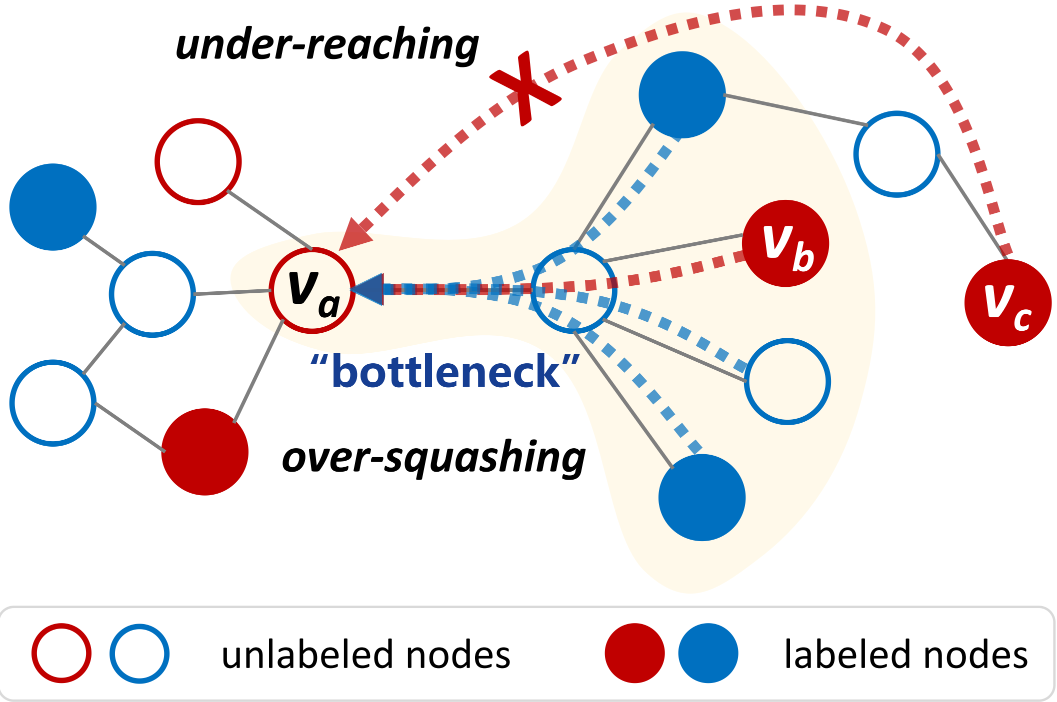


- **Measurements for modeling nodes relations**
  - node features/degrees/encodings, edge attributes, etc. **Node positions?**

Acknowledgement from https://arxiv.org/abs/2103.03036

Understanding Topology-imbalance

- **Q1: Why does topology-imbalance affect graph representation learning?**
- Q2: What kind of graphs are susceptible to topology-imbalance?



- **Under-reaching ➔** Reaching **C**oefficient *(RC)*

$$RC = \frac{1}{|\mathcal{V}_U|} \sum_{v_i \in \mathcal{V}_U} \frac{1}{|\mathcal{V}_L^{y_i}|} \sum_{v_j \in \mathcal{V}_L^{y_i}} \left(1 - \frac{\log |\mathcal{P}_{sp}(v_i, v_j)|}{\log D_{\mathcal{G}}}\right)$$

RC↑: better reachability (more shortcuts/paths)

- **Over-squashing ➔** Squashing **C**oefficient *(SC)*

$$SC = \frac{1}{|\mathcal{V}_U|} \sum_{v_i \in \mathcal{V}_U} \frac{1}{|\mathcal{N}_{y_i}(v_i)|} \sum_{v_j \in \mathcal{N}_{y_i}(v_i)} \frac{\sum_{e_{kt} \in \mathcal{P}_{sp}(v_i, v_j)} Ric(v_k, v_t)}{|\mathcal{P}_{sp}(v_i, v_j)|}$$

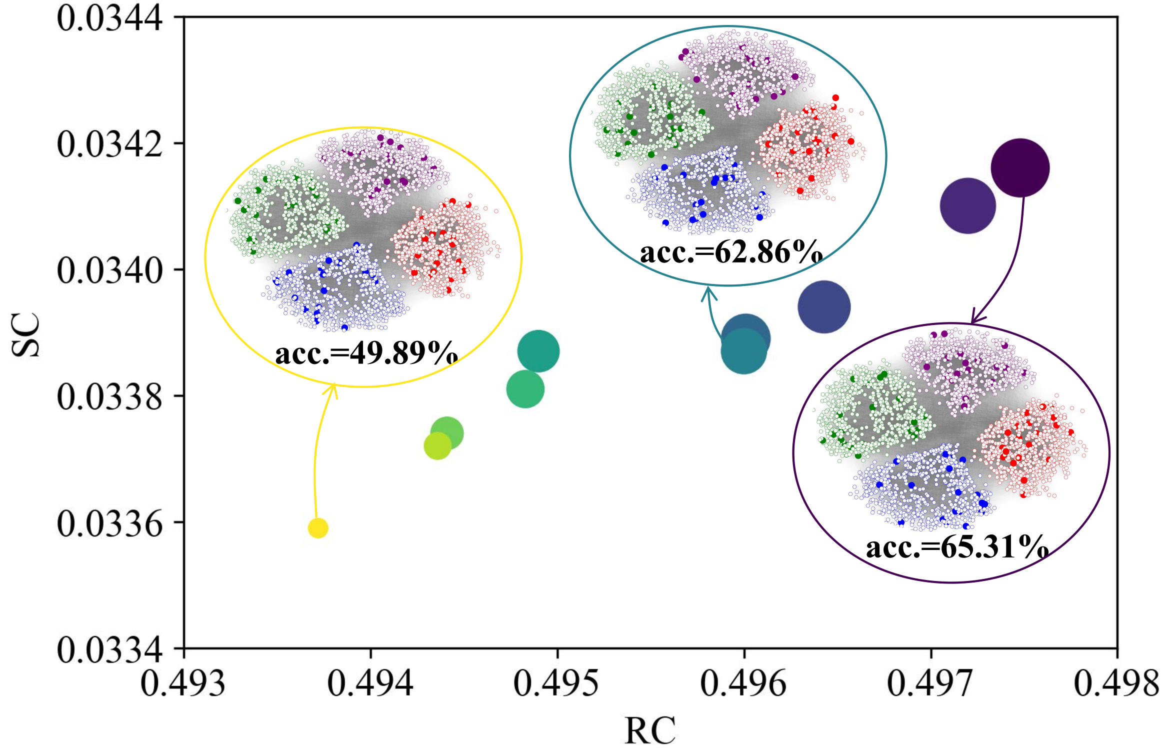
SC↑: lower squashing (more ring structures)

- **Conclusion: hurts the performance by under-reaching and over-squashing.**

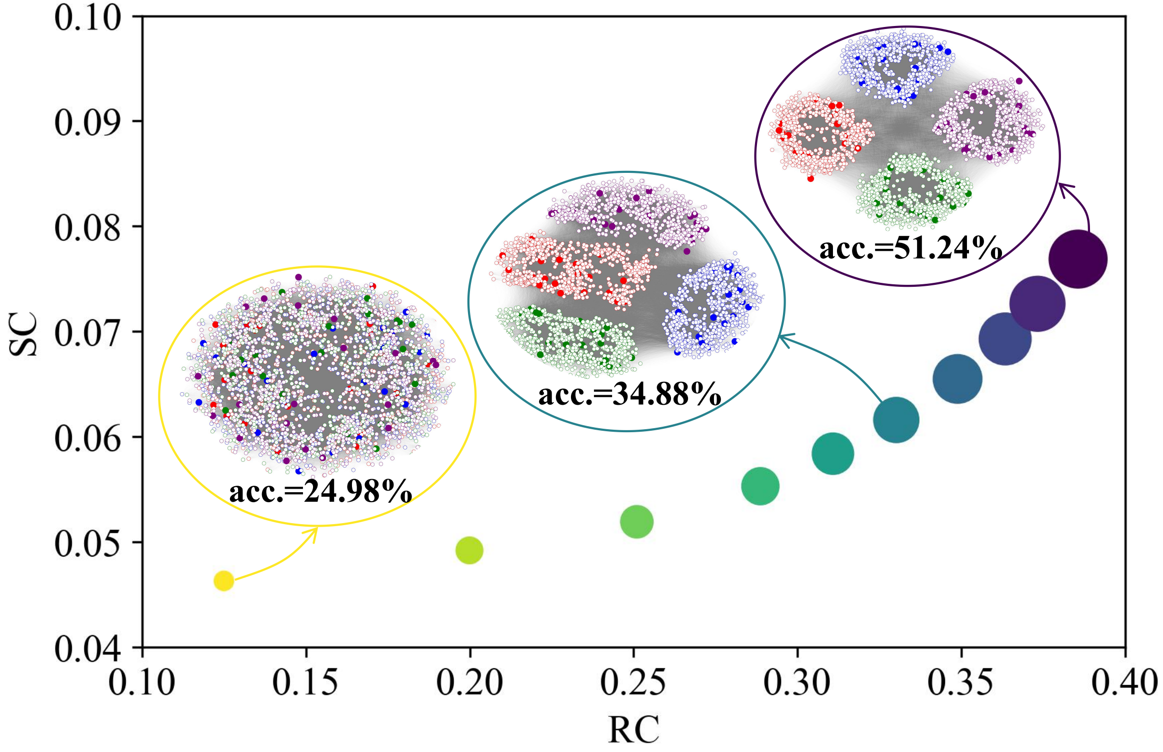
Understanding Topology-imbalance

- Q1: Why does topology-imbalance affect graph representation learning?
- **Q2: What kind of graphs are <u>susceptible</u> to topology-imbalance?** SBM $\mathcal{G}(N, C, p, q)$

same structure + **different** labeled nodes

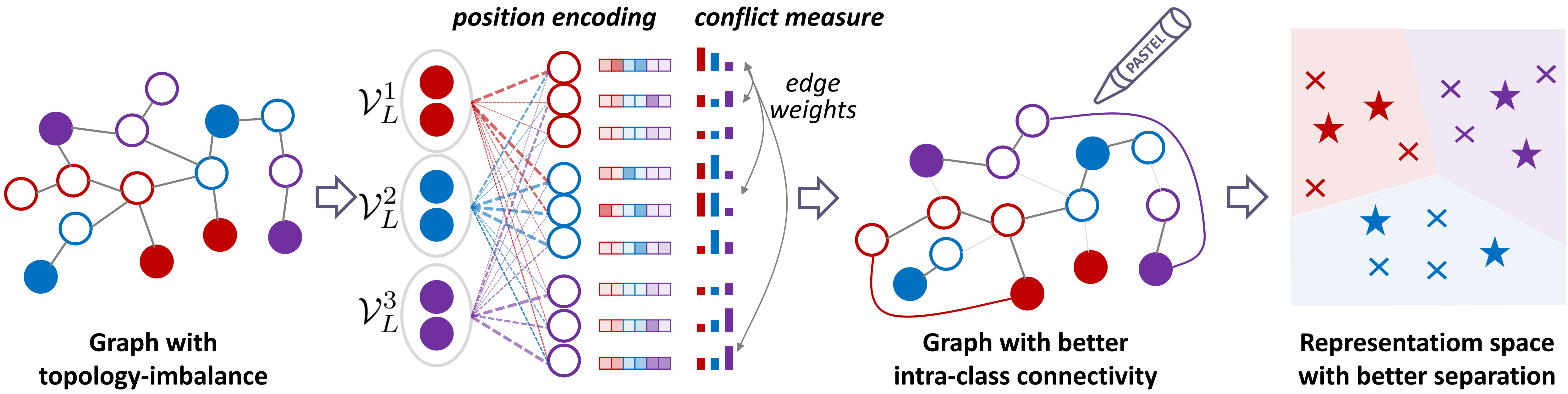


different structure + **same** labeled nodes



- **Conclusion: poor reachability (smaller *RC*) and stronger squashing (smaller *SC*)**

■ Position-Aware STructurE Learning framework (PASTEL)



- **Task: Semi-supervised node classification**
- **Position-aware Structure Learning:** **anchor-based** position encoding method
- **Class-wise Conflict Measure:** guide what nodes should be **more closely connected**
- **Learning with the Optimized Structure:** original + position + feature + constraints

Position-aware Structure Learning

Anchor-based Position Encoding

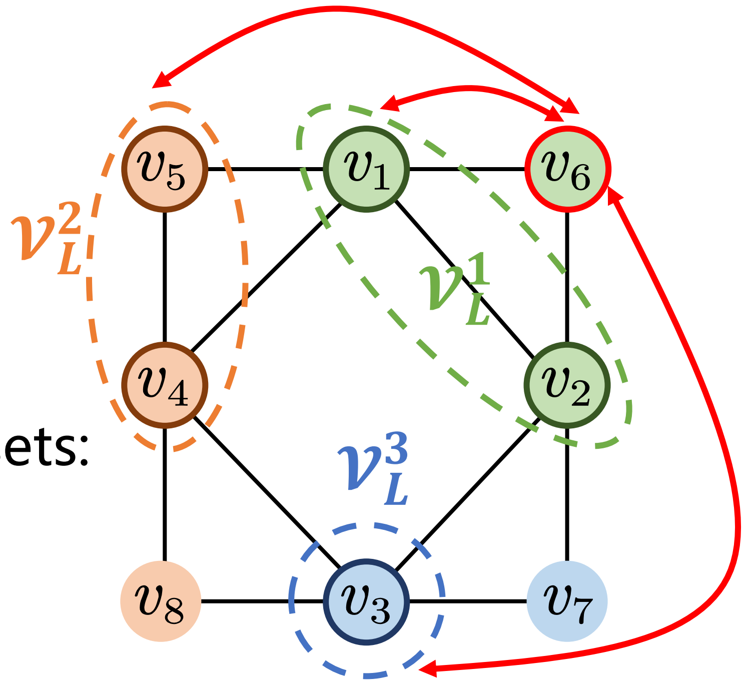
- separate labeled nodes **by class**: $\mathcal{V}_L = \{\mathcal{V}_L^1, \mathcal{V}_L^2, \cdots, \mathcal{V}_L^C\}$
- considerate an **unlabeled node v_i** (e.g. v_6):
  - measure **position relations** between v_i and anchor sets:

$$\phi(v_i, \mathcal{V}_L^c) = \frac{\sum_{v_j \in \mathcal{N}_c(v_i)} |\mathcal{P}_{sp}(v_i, v_j)|}{|\mathcal{N}_c(v_i)|}$$

  - **position-aware encoding** of node v_i:

$$\mathbf{p}_i = \left(\phi\left(v_i, \mathcal{V}_L^1\right), \phi\left(v_i, \mathcal{V}_L^2\right), \cdots, \phi\left(v_i, \mathcal{V}_L^C\right)\right)$$

  - transform to **learnable** vector: $\mathbf{h}_i^p = \mathbf{W}_\phi \cdot \mathbf{p}_i$



$$\mathcal{V}_L = \{\mathcal{V}_L^1, \mathcal{V}_L^2, \mathcal{V}_L^3\}$$

$$\mathbf{p}_6 = \left(\phi(v_6, \mathcal{V}_L^1), \phi(v_6, \mathcal{V}_L^2), \phi(v_6, \mathcal{V}_L^3)\right)$$

$$\mathbf{h}_6^p = \mathbf{W}_\phi \cdot \mathbf{p}_6$$

Position-aware Metric Learning

- consider both **feature info** and **position-based similarity** to form an edge:

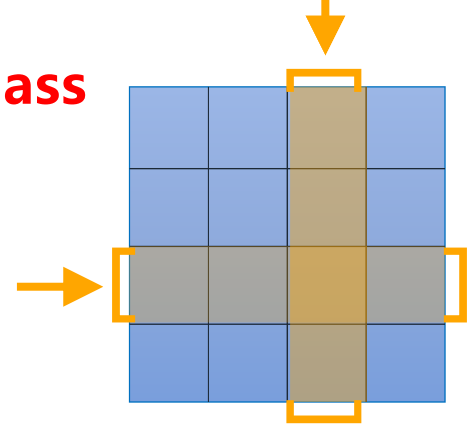
$$a_{ij}^P = \frac{1}{m} \sum_{h=1}^{m} \cos\left(\mathbf{W}_h \cdot \left(\mathbf{z}_i || \mathbf{h}_i^p\right), \mathbf{W}_h \cdot \left(\mathbf{z}_j || \mathbf{h}_j^p\right)\right)$$

Class-wise Conflict Measure

- **Group PageRank (GPR)**
  - traditional PageRank ➔ **Group** PageRank (label-aware): measure supervision information **from labeled nodes of each class**

$$\mathbf{P}^{gpr}(c) = (1-\alpha)\mathbf{A}'\mathbf{P}^{gpr}(c) + \alpha \mathbf{I}_c, \qquad \mathbf{P}^{gpr} = \alpha\left(\mathbf{E} - (1-\alpha)\mathbf{A}'\right)^{-1}\mathbf{I}^*$$

$$\mathbf{I}_c^i = \begin{cases} \frac{1}{|\mathcal{V}_L^c|}, & if\ y_i = c \\ 0, & otherwise \end{cases} \qquad \mathbf{I}^* = \{\mathbf{I}_c, c = 1, 2, \cdots, C\}$$



  - **Expecting:** GPR vector of nodes to form a **"sharp"** distribution focusing on their **ground truth label** *(see Experiments Sec.)*
- **Control the connection strength of an edge**
  - GPR vectors measure **conflict**: $\kappa_{ij} = \mathrm{KL}\left(\mathbf{P}_i^{gpr}, \mathbf{P}_i^{gpr}\right)$
  - conflict ➔ edge weight: $w_{ij} = \frac{1}{2}\left[-\cos\frac{\mathrm{Rank}(\kappa_{ij})}{|\mathcal{V}|\times|\mathcal{V}|} * \pi + 1\right]$, $\tilde{a}_{ij}^P = w_{ij}\cdot a_{ij}^P$ ➔ position-aware $\mathbf{A}_\mathbf{P}$

Learning with the Optimized Structure

Graph structure mixing and optimization

- **position-aware** adjacency: $\mathrm{A}_P = \{\tilde{a}_{ij}^{P}, i, j \in \{1, 2, \cdots, N\}\}$
- **node feature view** adjacency: $\mathrm{A}_N = \{a_{ij}^{N}, i, j \in \{1, 2, \cdots, N\}\}$,

$$a_{ij}^{N} = \frac{1}{m} \sum_{h=1}^{m} \cos\left(\mathbf{W}_h \cdot \left(\mathbf{x}_i || \mathbf{h}_i^{p_0}\right), \mathbf{W}_h \cdot \left(\mathbf{x}_j || \mathbf{h}_j^{p_0}\right)\right)$$

- **mixing**:

$$\mathrm{A}^* = \lambda_1 \mathrm{D}^{-\frac{1}{2}} \mathrm{A} \mathrm{D}^{-\frac{1}{2}} + (1 - \lambda_1)\left(\lambda_2 f\left(\mathrm{A}_N\right) + (1 - \lambda_2) f\left(\mathrm{A}_P\right)\right) \rightarrow \boxed{\mathcal{G}^* = (\mathrm{A}^*, \mathrm{X})}$$

Learning objectives

- structure quality control: $\underbrace{\mathcal{L}_{smooth} = \frac{1}{N^2} \mathrm{tr}\left(\mathrm{X}^T \mathrm{L}^* \mathrm{X}\right)}_{\textit{smooth}}$, $\underbrace{\mathcal{L}_{con} = \frac{1}{N} \mathbf{1}^T \log(\mathrm{A}^* \mathbf{1})}_{\textit{connectivity}}$, $\underbrace{\mathcal{L}_{spar} = \frac{1}{N^2} ||\mathrm{A}^*||_F^2}_{\textit{sparsity}}$

- classification loss: $\mathrm{Z} = \text{GNN-Encoder}(\mathrm{A}^*, \mathrm{X}), \hat{\mathcal{Y}} = \text{Classifier}(\mathrm{Z}), \mathcal{L}_{cls} = \text{Cross-Entropy}(\mathcal{Y}, \hat{\mathcal{Y}})$
- **overall loss**: $\mathcal{L} = \mathcal{L}_{cls} + \beta_1 \mathcal{L}_{smooth} + \beta_2 \mathcal{L}_{con} + \beta_3 \mathcal{L}_{spar}$

Experimental Setups

- **Datasets**
  - **real-world datasets:** Cora, Citeseer, Photo, Actor, Chameleon, Squirrel
  - **synthetic graph**: Stochastic Block Model (**SBM**)
- **Baselines**
  - **GNN backbones:** GCN, GAT, APPNP, GraphSAGE
  - **topology-imbalance specific baselines:** ReNode
  - **graph structure learning baselines:** DropEdge, AddEdge, SDRF, NeuralSparse, IDGL
- **Classification task setting:** set the number of **labeled nodes** in each class to be **20**
- **Metrics**
  - Weighted-F1 (W-F1) ± standard deviation
  - Macro-F1 (M-F1) ± standard deviation

Node Classification on Real-world Graphs

		Cora		Citeseer		Photo		Actor		Chameleon		Squirrel	
Backbone	Model	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1	W-F1	M-F1
GCN	original	79.4±0.9	77.5±1.5	66.3±1.3	62.2±1.2	85.4±2.8	84.6±1.3	21.8±1.3	20.9±1.4	30.5±3.4	30.5±3.3	21.9±1.2	21.9±1.2
	ReNode	80.0±0.7	78.4±1.3	66.4±1.0	62.4±1.1	86.2±2.4	85.3±1.6	21.2±1.2	20.2±1.6	30.3±3.2	30.4±2.8	22.4±1.1	22.4±1.1
	AddEdge	79.0±0.9	77.0±1.4	66.2±1.3	62.2±1.3	85.5±1.5	86.1±1.8	21.2±1.3	20.3±1.5	30.6±1.6	30.4±1.7	21.7±1.5	21.7±1.5
	DropEdge	79.8±0.8	77.8±1.0	66.6±1.4	63.4±1.6	86.8±1.7	85.4±1.3	22.4±1.0	21.4±1.3	30.6±3.5	30.6±3.3	22.8±1.2	22.8±1.2
	SDRF	82.1±0.8	80.6±0.8	69.6±0.4	66.6±0.3	> 5 days	> 5 days	> 5 days	> 5 days	39.1±1.2	39.0±1.2	> 5 days	> 5 days
	NeuralSparse	81.7±1.4	80.9±1.4	71.8±1.2	69.0±1.0	89.7±1.9	88.7±1.8	24.4±1.5	23.6±1.6	44.9±3.0	44.9±2.8	28.1±1.8	28.1±1.8
	IDGL	82.3±0.6	81.0±0.9	71.7±1.0	68.0±1.3	88.6±2.3	88.8±1.4	24.9±0.8	22.0±0.7	55.4±1.8	55.0±1.7	28.8±2.3	28.9±2.2
	PASTEL	**82.5±0.3**	**81.2±0.3**	**72.9±0.8**	**69.3±0.9**	**91.4±2.7**	**91.3±2.2**	**26.4±1.0**	**24.4±1.2**	**57.8±2.4**	**57.3±2.4**	**37.5±0.6**	**37.5±0.7**
GAT	original	78.3±1.5	76.4±1.7	64.4±1.7	60.6±1.7	88.2±2.9	86.2±2.6	21.8±1.2	20.9±1.1	29.9±3.5	29.9±3.1	20.5±1.4	20.5±1.4
	ReNode	78.9±1.2	77.2±1.5	64.9±1.6	61.0±1.5	89.1±2.4	87.1±2.6	21.5±1.2	20.5±1.1	29.2±2.3	29.1±2.0	20.4±1.8	20.4±1.8
	AddEdge	78.0±1.6	76.2±1.6	64.0±1.3	60.2±1.3	88.2±2.4	86.2±2.5	21.3±1.2	20.3±1.1	29.8±1.7	29.6±1.5	20.7±1.6	20.7±1.6
	DropEdge	78.7±1.3	76.9±1.5	64.5±1.4	60.5±1.3	88.9±1.9	87.1±2.1	22.9±1.2	21.8±1.1	30.3±1.6	30.2±1.2	21.2±1.5	21.2±1.5
	SDRF	77.9±0.7	75.9±0.9	64.9±0.6	61.9±0.9	> 5 days	> 5 days	> 5 days	> 5 days	43.0±1.9	42.5±1.9	> 5 days	> 5 days
	NerualSparse	81.4±4.8	79.4±4.8	64.8±1.5	61.9±1.3	90.2±2.5	88.0±2.3	23.4±1.7	**22.4±1.5**	45.6±2.1	45.5±1.8	28.8±1.3	28.8±1.3
	IDGL	80.6±1.0	79.7±0.9	66.5±1.5	61.9±1.9	89.9±3.1	87.7±2.6	22.4±1.5	21.8±1.2	48.4±4.0	47.8±3.1	27.0±2.6	27.0±2.6
	PASTEL	**81.9±1.4**	**80.7±1.2**	**66.6±1.9**	**62.0±1.7**	**91.8±3.2**	**89.4±2.9**	**24.4±2.6**	22.1±2.6	**52.1±2.7**	**52.5±2.8**	**35.3±0.9**	**35.3±0.8**
APPNP	original	80.6±1.6	79.3±1.2	66.5±1.5	62.3±1.5	89.3±1.6	86.3±1.7	21.1±1.5	20.7±1.1	35.3±4.0	35.0±3.8	23.1±1.6	23.1±1.6
	ReNode	81.1±0.9	79.9±0.9	66.6±1.7	62.4±1.6	89.6±1.4	87.2±1.3	20.2±2.0	20.0±1.7	33.5±2.5	33.3±2.3	23.9±2.0	23.9±2.0
	AddEdge	80.3±1.3	78.8±1.1	66.6±2.1	62.5±2.1	89.3±1.2	86.4±1.2	21.5±1.3	20.7±1.4	35.7±1.7	35.4±1.2	23.1±1.6	23.2±1.7
	DropEdge	80.9±1.4	79.4±1.2	66.7±2.0	63.0±1.9	90.0±1.2	87.0±1.2	21.8±1.8	20.8±1.4	36.0±1.7	35.7±1.6	23.3±1.7	23.3±1.7
	SDRF	80.7±0.9	79.1±0.8	67.1±0.6	63.1±0.8	> 5 days	> 5 days	> 5 days	> 5 days	36.5±2.1	35.8±2.1	> 5 days	> 5 days
	NerualSparse	81.1±1.4	79.9±1.2	66.8±1.9	62.7±1.9	91.3±1.8	89.4±1.6	21.8±1.9	**21.4±1.5**	39.1±2.9	38.7±2.8	28.3±1.5	28.3±1.5
	IDGL	81.3±0.9	**80.2±0.9**	67.0±1.3	62.9±1.3	91.6±1.3	88.6±2.2	21.4±2.4	20.1±2.4	41.2±2.2	40.6±2.6	29.6±2.3	29.7±2.2
	PASTEL	**82.0±1.0**	80.0±0.9	**67.3±1.3**	**63.2±1.5**	**92.3±3.1**	**89.9±2.5**	**22.5±2.0**	20.9±2.1	**44.2±3.2**	**43.8±3.4**	**34.6±1.6**	**34.6±1.6**
GraphSAGE	original	75.4±1.6	74.1±1.6	64.8±1.6	60.7±1.6	86.1±2.5	83.3±2.4	24.0±1.2	23.2±1.0	36.5±1.6	36.2±1.6	27.2±1.7	27.2±1.7
	ReNode	76.4±0.9	75.0±1.1	65.4±1.7	61.2±1.7	86.5±1.7	84.1±1.7	23.7±1.2	22.8±1.0	36.4±1.9	36.1±1.9	27.7±1.8	27.7±1.8
	AddEdge	75.2±1.2	73.7±1.2	65.0±1.4	60.9±1.3	86.1±2.8	83.4±2.6	23.8±1.7	23.2±1.6	36.5±1.5	36.2±1.3	26.9±2.1	26.9±2.1
	DropEdge	76.0±1.6	74.5±1.6	65.1±1.4	60.9±1.4	86.2±1.6	83.5±1.4	24.1±1.0	23.3±0.9	37.5±1.4	37.2±1.4	27.5±1.8	27.5±1.8
	SDRF	75.7±0.8	74.6±0.8	65.3±0.6	**61.4±0.6**	> 5 days	> 5 days	> 5 days	> 5 days	41.5±2.6	41.6±2.7	> 5 days	> 5 days
	NerualSparse	79.7±1.8	77.8±1.6	64.7±1.4	61.1±1.3	89.1±5.4	86.7±5.5	25.1±1.2	**24.4±1.1**	39.1±1.9	39.0±1.9	32.2±2.4	32.2±2.4
	IDGL	79.2±0.9	78.4±0.8	65.6±0.9	61.3±1.2	90.0±1.0	86.3±1.3	24.0±2.6	22.4±2.7	43.8±3.4	43.0±3.2	33.9±0.9	33.9±0.8
	PASTEL	**81.1±0.8**	**79.8±0.7**	**65.7±1.1**	**61.4±1.4**	**92.0±0.6**	**89.0±1.0**	**26.0±2.4**	23.6±2.7	**47.7±0.9**	**46.9±0.9**	**35.5±1.4**	**35.5±1.4**

- **PASTEL shows overwhelming superiority in improving the performance of backbones on all datasets.**

■ *Node Classification on Cora with different imbalance level*

- Cora-L, Cora-M, Cora-H: topology-imbalance level low(L)/medium(M)/high(H)
- GNN backbone: GCN

	Cora-L		Cora-M		Cora-H	
	RC	*SC*	*RC*	*SC*	*RC*	*SC*
	0.4130	-0.6183	0.4100	-0.6204	0.4060	-0.6302
	W-F1 (%)	Δ (%)	W-F1 (%)	Δ (%)	W-F1 (%)	Δ (%)
GCN	80.9±0.9	—	78.8±0.8	—	77.5±1.0	—
ReNode	81.3±0.7	↑0.4	79.3±0.8	↑0.5	78.3±1.1	↑0.8
SDRF	81.0±0.7	↑0.1	78.9±0.8	↑0.1	77.9±0.7	↑0.4
IDGL	82.5±1.0	↑1.6	80.4±1.0	↑1.6	81.6±1.1	↑4.1
PASTEL	**82.7±0.9**	**↑1.8**	**81.0±0.9**	**↑2.2**	**81.9±1.1**	**↑4.4**

- PASTEL performs best on all datasets with different imbalance level.
- achieve up to 4.4% improvement on the highly topology-imbalance dataset.

Node Classification on Synthetic Graphs

- **Stochastic Block Model (SBM)** $\mathcal{G}(N, C, p, q)$ (*N*=3000, *C*=6)
- **GNN backbone: GCN**

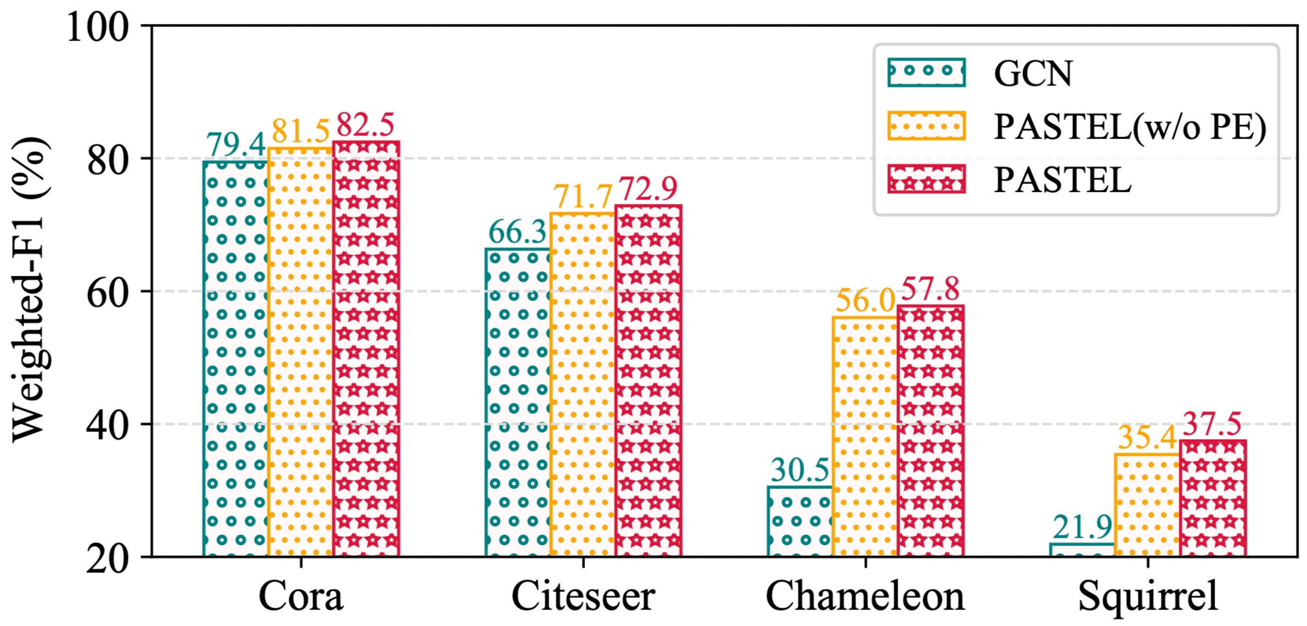
topology-imbalance level: high → low

	SBM-1		SBM-2		SBM-3		SBM-4		SBM-5		SBM-6		SBM-7	
p	0.5000		0.5000		0.5000		0.5000		0.5000		0.5000		0.5000	
q	0.0300		0.0100		0.0083		0.0071		0.0063		0.0056		0.0050	
RC	0.4979		0.4984		0.4990		0.4994		0.5002		0.5004		0.5009	
SC	0.0998		0.0999		0.1000		0.1001		0.1007		0.1017		0.1144	
	W-F1	Δ	W-F1	Δ	W-F1	Δ	W-F1	Δ	W-F1	Δ	W-F1	Δ	W-F1	Δ
GCN	40.29	—	42.37	—	42.99	—	44.13	—	45.19	—	45.21	—	45.22	—
ReNode	41.33	↑1.04	42.40	↑0.03	43.21	↑0.22	44.56	↑0.43	45.20	↑0.01	45.08	↓0.13	44.89	↓0.33
PASTEL	**45.67**	**↑5.38**	**57.61**	**↑15.24**	**58.33**	**↑15.34**	**60.29**	**↑16.16**	**66.41**	**↑21.22**	**66.45**	**↑21.24**	**66.57**	**↑21.35**

- **PASTEL can increase the classification Weighted-F1 score by 5.38%-21.35% on SBM graphs with different community structures, showing superior effectiveness.**

Ablation Study: Impact of the Position Encoding

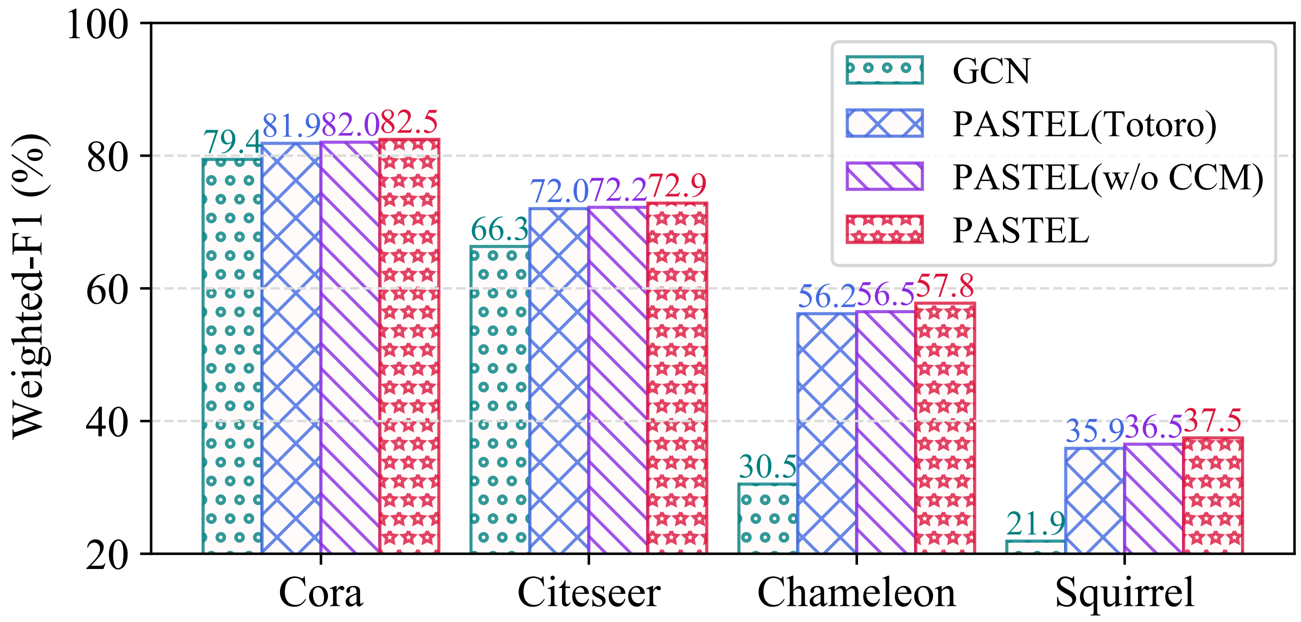
- **Variant: PASTEL (w/o PE) with GCN as backbone**
  - removes the position encoding module
  - directly take the node features for metric learning



- **The structure learning strategy of PASTEL contributes the most.**
- **The position encoding still benefits learning better structure.**

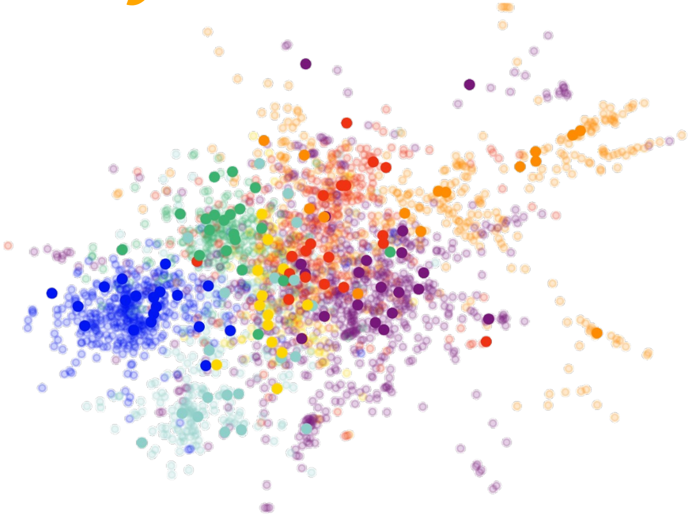
Ablation Study: Impact of the Class-wise Conflict Measure

- **Variant: PASTEL (w/o CCM) with GCN as backbone**
  - remove the class-wise conflict measure, directly takes the learned edge possibilities
- **Variant: PASTEL (Totoro) with GCN as backbone**
  - take the Totoro metric introduced in ReNode as the conflict measure of nodes

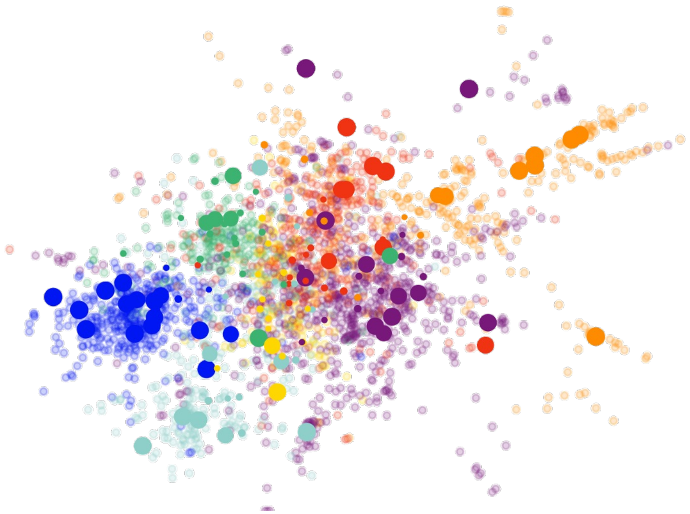


- **PASTEL consistently outperforms the other two variants.**

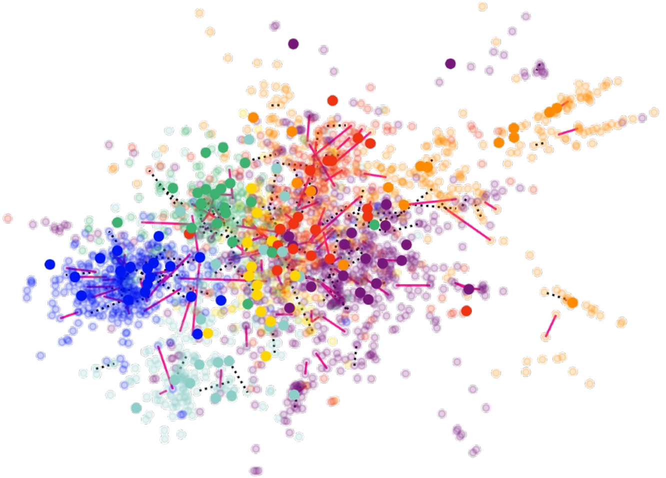
Analysis of Learned Structure: Visualization of Cora



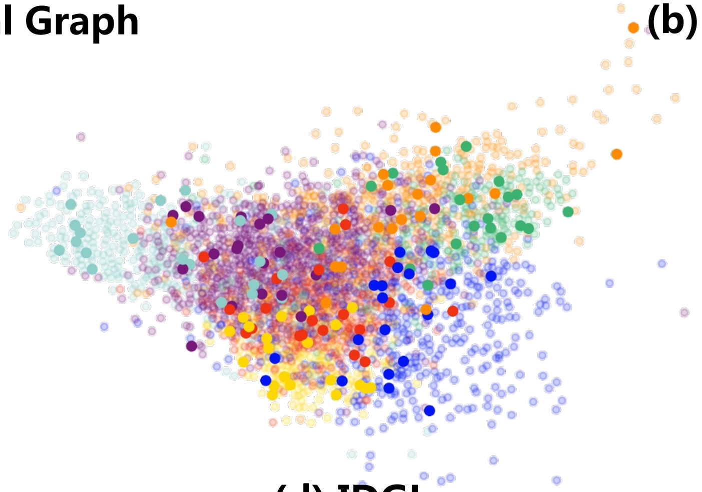
(a) Original Graph



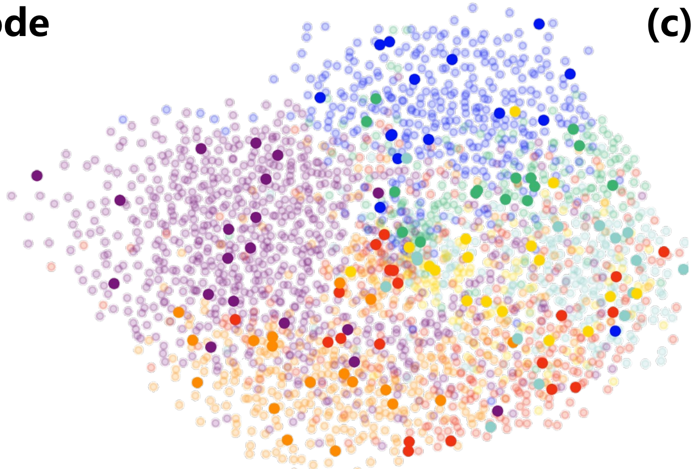
(b) ReNode



(c) SDRF



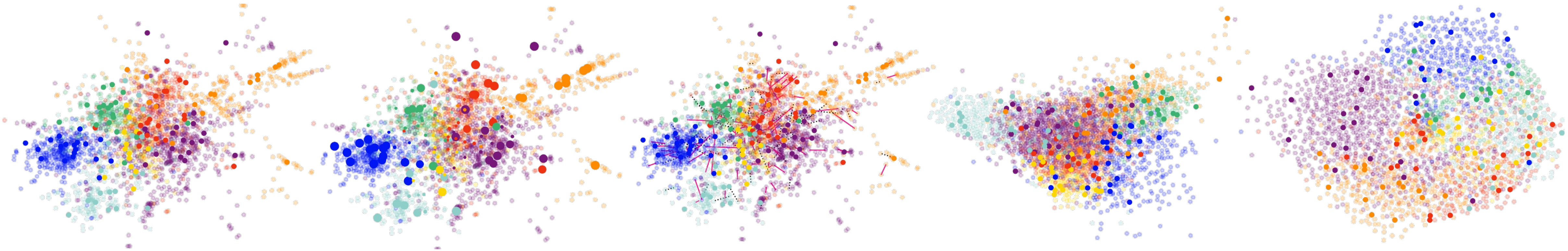
(d) IDGL



(e) PASTEL (Ours)

- **PASTEL can obtain graph structure with clearer class boundaries.**

Analysis of Learned Structure: Change of RC and SC



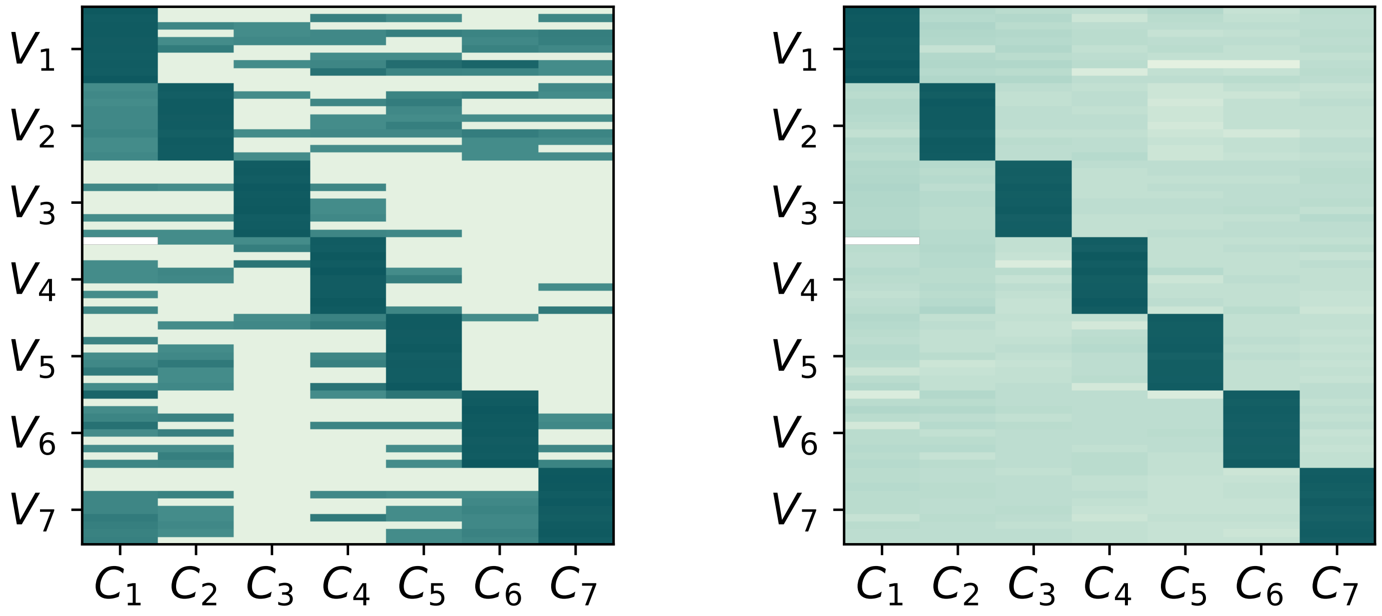
(a) Original Graph (b) ReNode (c) SDRF (d) IDGL *(e) PASTEL (Ours)*

	Original Graph	ReNode	SDRF	IDGL	**PASTEL**
RC	0.4022	0.4022	0.4686	0.5028	0.5475
SC	-0.6299	-0.6299	-0.4942	-0.4069	-0.3389
W-F1 (%)	79.44	80.34	82.01	82.38	**82.86**

- **All the structure learning methods learn structures with larger RC & SC, leading the performance improvement of node classification.**

Analysis of Learned Structure: Change of GPR Vector

- randomly choose **10** nodes for **each class** in Cora
- visualize their GPR vectors on **the original graph** and **the learned graph**



(a) Original Graph **(b) Learned Graph**

- V_i : 10 nodes of class i
- C_i : the i-th class

- **the class-wise conflict measure plays an important role on giving guidance for more class connectivity orthogonality.**

Conclusion and Future Works

- We provide **a new understanding** and **two metrics** of topology-imbalance in the perspective of **under-reaching** and **over-squashing**, answering:
  - how topology-imbalance affects GNN’s performance
  - what graphs are susceptible to topology-imbalance
- PASTEL designs an **anchor-based position encoding mechanism** and a **class-wise conflict measure** to obtain structures with **better in-class connectivity**.
- Future works: incorporate the proposed two quantitative metrics **into the learning process** to address topology-imbalance more directly.

Position-aware Structure Learning for Graph Topology-imbalance by Relieving Under-reaching and Over-squashing

Qingyun Sun
Beihang University
Beijing, China
sunqy@buaa.edu.cn

Jianxin Li
Beihang University
Beijing, China
lijx@buaa.edu.cn

Haonan Yuan
Beihang University
Beijing, China
yuanhn@act.buaa.edu.cn

Xingcheng Fu
Beihang University
Beijing, China
fuxc@act.buaa.edu.cn

Hao Peng
Beihang University
Beijing, China
penghao@act.buaa.edu.cn

Cheng Ji
Beihang University
Beijing, China
jicheng@act.buaa.edu.cn

Qian Li
Beihang University
Beijing, China
liqian@act.buaa.edu.cn

Philip S. Yu
University of Illinois Chicago
Chicago, USA
psyu@uic.edu

paper: https://arxiv.org/pdf/2208.08302.pdf
code: https://github.com/RingBDStack/PASTEL

2022.10.18