NeurIPS 2023 GLFrontiers Workshop

Non-backtracking Graph Neural Networks

<u>Seonghyun Park</u>[1†], Narae Ryu[2†], Gahee Kim[2], Dongyeop Woo[1], Se-Young Yun[2‡], Sungsoo Ahn[1‡]

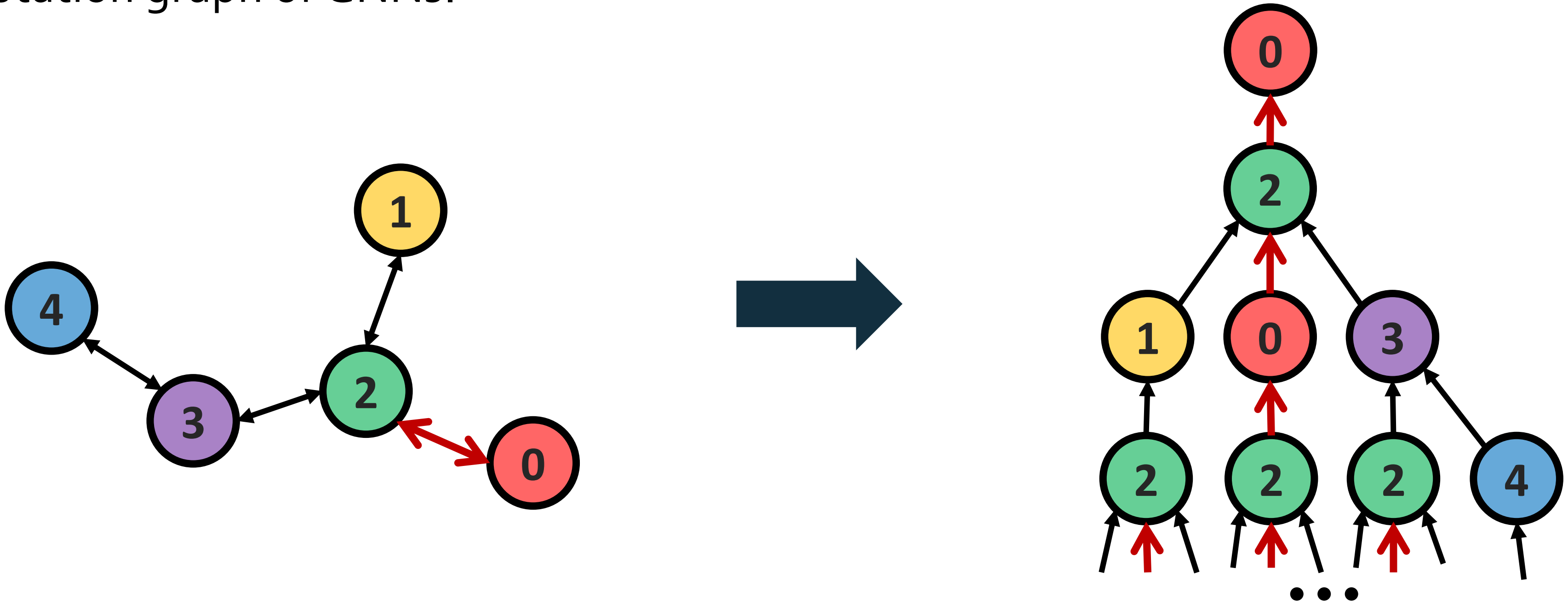
[1]POSTECH [2]KAIST

2023/12/15

†: Equal contribution, ‡: Co-corresponding author

Motivation – redundancy in nodes

- GNNs suffer **over-squashing**, the phenomenon of exponentially growing information **squashed** into a fixed-sized representation[1].
- To **reduce** the exponentially growing information, we aim to remove **redundant nodes** in the computation graph of GNNs.



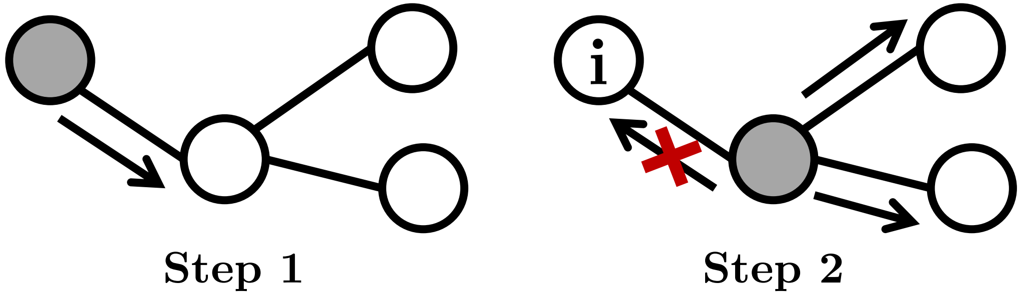
Graph G

Computation graph for *node* 0 of conventional GNNs

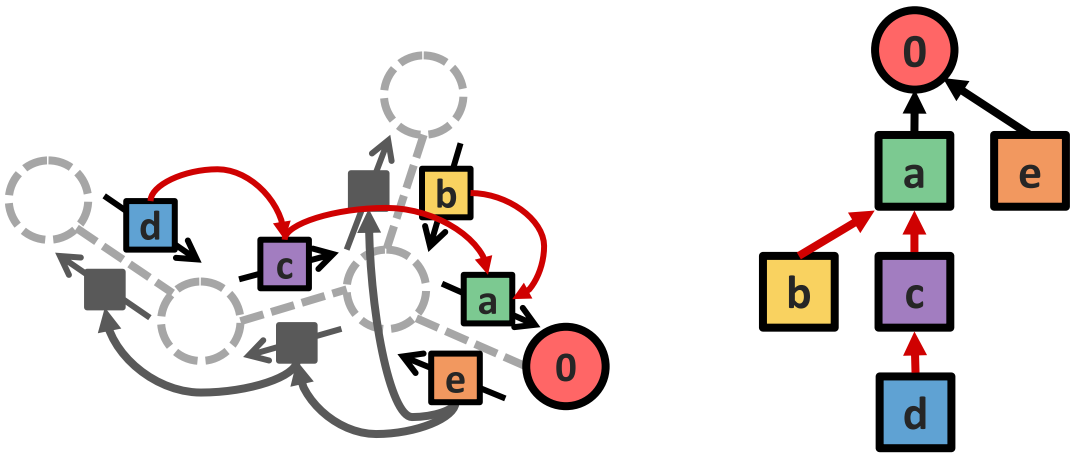
[1] One the bottlenecks of GNNs and Its Pratical Implications *(ICLR 2021)*

Motivation – redundancy in nodes

- With **non-backtracking**[2], we remove the *redundant nodes* in the computation graph.
  - **Non-backtracking:** preventing re-visits to previous nodes it came from.
- From a high-level, we do message-passing on **edge features** with ***neighbors*** selected by **non-backtracking.**



Example of non-backtracking
Preventing re-visits to previous nodes it came from

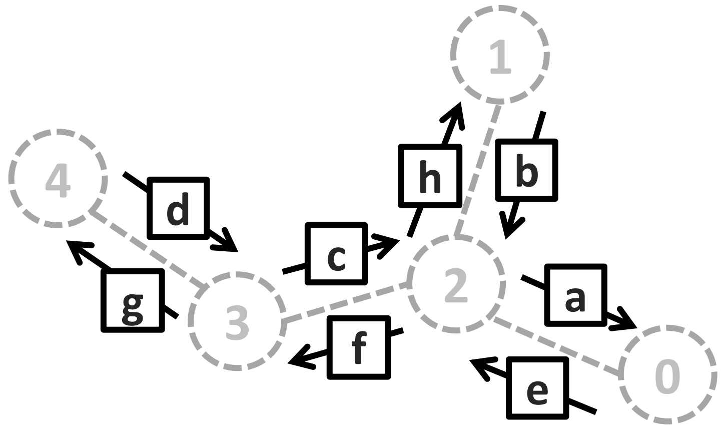


Message passing on edge features with non-backtracking

[1] LGNN, Redundancy-free message passing for graph neural networks *(NeurIPS 2022)*
[2] Supervised community detection with line graph neural networks *(ICLR 2019)*

<u>N</u>on-<u>BA</u>cktracking GNNs (NBA-GNNs)

1) Construct **edge-wise features $\boldsymbol{h}_{\boldsymbol{i}\to\boldsymbol{j}}$** for each edge $i \to j$, considering **directions.**



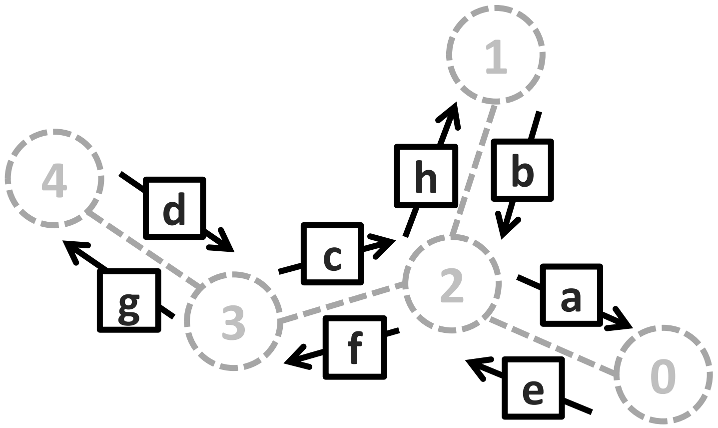
1) Edge-wise features considering directions

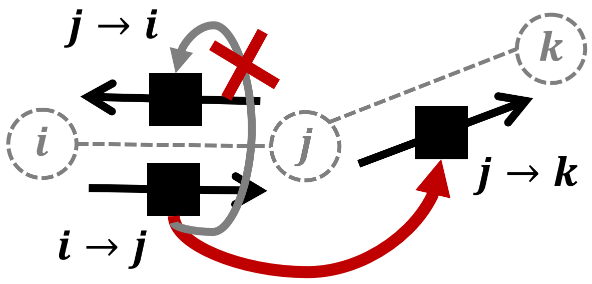
Non-BAcktracking GNNs (NBA-GNNs)

1) Construct **edge-wise features $\boldsymbol{h}_{i\to j}$** for each edge $i \to j$, considering **directions.**

2) **Update edge features $\boldsymbol{h}_{i\to j}$** based on the non-backtracking, using a backbone GNN.

$$\underbrace{h_{i\to j}}_{\text{Updated feature}} = \mathrm{UPDATE}\left(h_{i\to j}, \mathrm{AGG}\left(\left\{ h_{j\to k} : k \in \underbrace{\mathcal{N}(j) \setminus i}_{\text{Neighbors selected by Non-backtracking}} \right\} \right)\right)$$





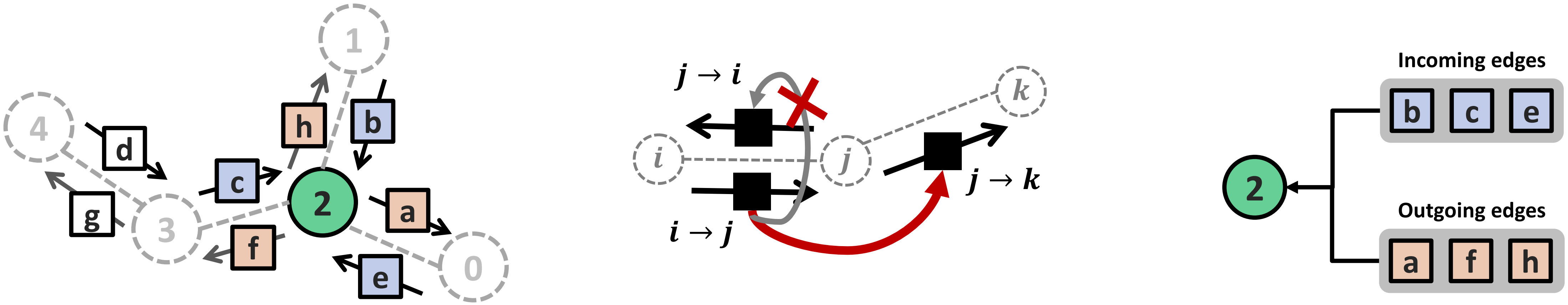
1) Edge-wise features considering directions 2) Update feature based on non-backtracking

Non-BAcktracking GNNs (NBA-GNNs)

1) Construct **edge-wise features $\boldsymbol{h}_{i \to j}$** for each edge $i \to j$, considering **directions.**

2) **Update edge features $\boldsymbol{h}_{i \to j}$** based on the **non-backtracking**, using a backbone GNN.

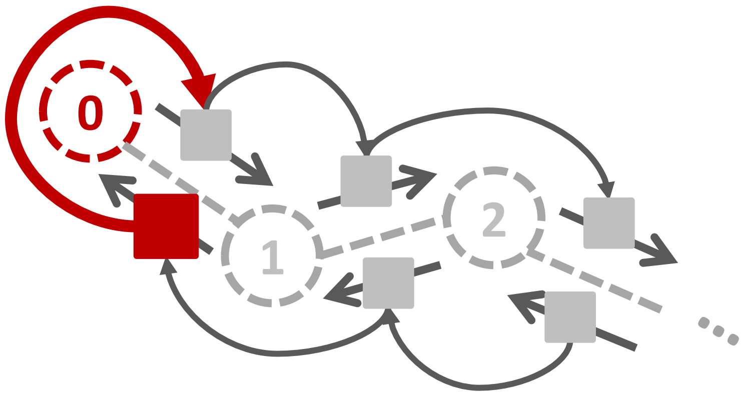
3) Compute node-wise representations $\boldsymbol{h}_i$, based on incoming & outgoing edges.



1) Edge-wise features considering directions **2) Update feature based on non-backtracking** **3) Computation graph for node 2**

Begrudgingly backtracking

- **Drawbacks** of non-backtracking
  - **Dangling nodes**: nodes with only one neighbor, e.g., **node 0**.
  - The *information gets* ***stuck***, failing to be propagated.



- **Begrudgingly backtracking**[1]
  - In these cases backtracking updates are allowed, e.g., updating $h_{0\to1}$ with $h_{1\to0}$.

[1] Faster clustering via non-backtracking random walks

Theoretical analyses

- **Thm. 1** NBA-GNN **mitigates over-squashing**.
  - **Sensitivity** has been used to assess *over-squashing*, where *higher sensitivity results less over-squashing*[1].
  - We show the *sensitivity upper bound for NBA-GNN* is **larger** than that of conventional GNNs.

Sensitivity $\left\|\frac{\partial h_j}{\partial x_i}\right\|$ $\leq$ **Conventional GNNs** $\leq$ **NBA-GNNs**

- **Thm. 2** NBA-GNN **enhances the expressivity power**.
  - Non-backtracking inhibits a **spectral separation property** for *vertex community detections*[2].
  - We used this to analyze that NBA-GNN can **distinguish graphs generated by ER / SBM**[3].

[1] Understanding over-squashing and bottlenecks on graphs via curvature *(ICLR 2022)*
[2] Non-backtracking spectra of weighted inhomogeneous random graphs *(Mathematical and Statistic and Learning 2022)*
[3] Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs *(IEEE 2015)*

Experiment results

- **Long-range graph benchmark (lrgb)**
  - Graph datas where over-squashing is likely to happen.

Model	Peptides-func		Peptides-struct		PascalVOC-SP	
	AP ↑	Imp.	MAE ↓	Imp.	F1 ↑	Imp.
GCN	0.5930 ± 0.0023		0.3496 ± 0.0013		0.1268 ± 0.0060	
+ NBA	0.6951 ± 0.0024	+17%	0.2656 ± 0.0009	+22%	0.2537 ± 0.0054	+100%
+ NBA+LapPE	**0.7206** ± **0.0028**	+22%	**0.2472** ± **0.0008**	+29%	**0.3005** ± **0.0010**	+137%
GIN	0.5498 ± 0.0079		0.3547 ± 0.0045		0.1265 ± 0.0076	
+ NBA	0.6961 ± 0.0045	+27%	0.2534 ± 0.0025	+29%	0.3040 ± 0.0119	+140%
+ NBA+LapPE	**0.7071** ± **0.0067**	+29%	**0.2424** ± **0.0010**	+32%	**0.3223** ± **0.0010**	+155%
GatedGCN	0.5864 ± 0.0077		0.3420 ± 0.0013		0.2873 ± 0.0219	
+ NBA	0.6429 ± 0.0062	+10%	0.2539 ± 0.0011	+26%	0.3910 ± 0.0010	+36%
+ NBA+LapPE	**0.6982** ± **0.0014**	+19%	**0.2466** ± **0.0012**	+28%	**0.3969** ± **0.0027**	+38%

Show improvements regardless of the backbone GNN

Experiment results (cont.)

First, Second, Third

Method	Model	Peptides-func AP ↑	Peptides-struct MAE ↓	VOC-SP F1 ↑
GNNs	GCN	0.5930 ± 0.0023	0.3496 ± 0.0013	0.1268 ± 0.0060
	GIN	0.5498 ± 0.0079	0.3547 ± 0.0045	0.1265 ± 0.0076
	GatedGCN	0.5864 ± 0.0077	0.3420 ± 0.0013	0.2873 ± 0.0219
	GatedGCN+PE	0.6069 ± 0.0035	0.3357 ± 0.0006	0.2860 ± 0.0085
Subgraph GNNs	MixHop-GCN	0.6592 ± 0.0036	0.2921 ± 0.0023	0.2506 ± 0.0133
	MixHop-GCN+LapPE	0.6843 ± 0.0049	0.2614 ± 0.0023	0.2218 ± 0.0174
	PathNN	0.6816 ± 0.0026	0.2545 ± 0.0032	-
	CIN++	0.6569 ± 0.0117	0.2523 ± 0.0013	-
Transformers	Transformer+LapPE	0.6326 ± 0.0126	0.2529 ± 0.0016	0.2694 ± 0.0098
	GraphGPS+LapPE	0.6535 ± 0.0041	0.2500 ± 0.0005	**0.3748 ± 0.0109**
	SAN+LapPE	0.6384 ± 0.0121	0.2683 ± 0.0043	0.3230 ± 0.0039
	Exphormer	0.6527 ± 0.0043	0.2481 ± 0.0007	**0.3966 ± 0.0027**
	Graph MLP-Mixer/ViT	0.6970 ± 0.0080	**0.2449 ± 0.0016**	-
Rewiring methods	DIGL+MPNN	0.6469 ± 0.0019	0.3173 ± 0.0007	0.2824 ± 0.0039
	DIGL+MPNN+LapPE	0.6830 ± 0.0026	0.2616 ± 0.0018	0.2921 ± 0.0038
	DRew-GCN+LapPE	**0.7150 ± 0.0044**	0.2536 ± 0.0015	0.1851 ± 0.0092
	DRew-GIN+LapPE	**0.7126 ± 0.0045**	0.2606 ± 0.0014	0.2692 ± 0.0059
	DRew-GatedGCN+LapPE	0.6977 ± 0.0026	0.2539 ± 0.0007	0.3314 ± 0.0024
NBA-GNNs (Ours)	NBA-GCN	0.6951 ± 0.0024	0.2656 ± 0.0009	0.2537 ± 0.0054
	NBA-GCN+LapPE	**0.7207 ± 0.0028**	**0.2472 ± 0.0008**	0.3005 ± 0.0010
	NBA-GIN	0.6961 ± 0.0045	0.2775 ± 0.0057	0.3040 ± 0.0119
	NBA-GIN+LapPE	**0.7071 ± 0.0067**	**0.2424 ± 0.0010**	0.3223 ± 0.0063
	NBA-GatedGCN	0.6429 ± 0.0062	0.2539 ± 0.0011	**0.3910 ± 0.0010**
	NBA-GatedGCN+LapPE	0.6982 ± 0.0014	**0.2466 ± 0.0012**	**0.3969 ± 0.0027**

Show competitive results against baselines

Thank you

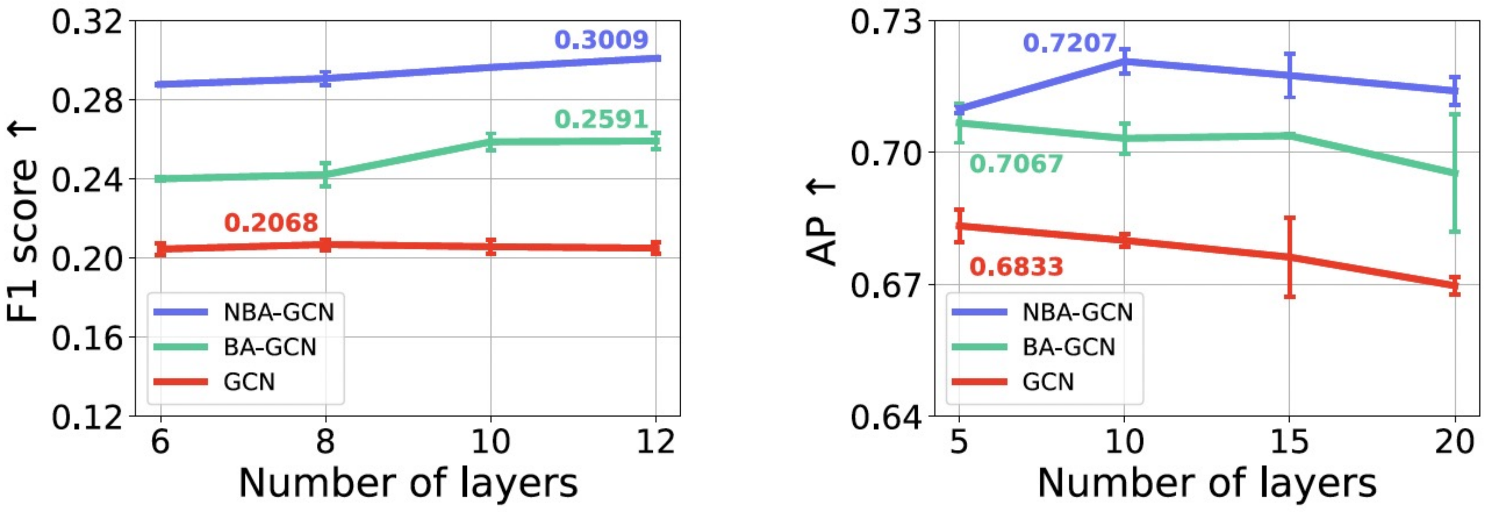
Appendix

A. Ablation study – non-backtracking

Q. Is **non-backtracking** really effective?

A. We compare "**non-backtracking**" & "**backtracking**" in ***PascalVOC-SP, peptides-func*** dataset

- Backtracking: update $h_{i\to j}^{t+1}$ using $h_{j\to i}^{t}$



Regarldess of the number of layers, **non-backtracking outperforms backtracking**

B. Ablation study – begrudgingly backtracking

Q. Is **begrudgingly** backtracking really effective?

A. We verify the effectiveness of begrudgingly(BG.) in ***peptides-func*** dataset

- Pascal-VOCSP does not have dangling nodes

Model	BG.	Peptides-func AP ↑
GCN	✗	0.7015 ± 0.0009
	✓	$\mathbf{0.7207 \pm 0.0028}$
GIN	✗	0.6825 ± 0.0075
	✓	$\mathbf{0.7071 \pm 0.0067}$
GatedGCN	✗	0.6710 ± 0.0009
	✓	$\mathbf{0.6982 \pm 0.0014}$

Begrudgingly backtracking consistently improves performance

C. Experiment – Transductive node classification

- Experiments on citation networks and heterophilic graphs

Model	Cora	CiteSeer	PubMed	Texas	Wisconsin	Cornell
GCN	0.8658±0.0060	0.7532±0.0134	0.8825±0.0042	0.6162±0.0634	0.6059±0.0438	0.5946±0.0662
+NBA	**0.8722±0.0095**	0.7585±0.0175	0.8826±0.0044	**0.7108±0.0796**	**0.7471±0.0386**	0.6108±0.0614
+NBA+LapPE	0.8720±0.0129	**0.7609±0.0186**	**0.8827±0.0048**	0.6811±0.0595	**0.7471±0.0466**	**0.6378±0.0317**
GraphSAGE	0.8632±0.0158	0.7559±0.0161	0.8864±0.0030	0.7108±0.0556	0.7706±0.0403	0.6027±0.0625
+NBA	**0.8702±0.0083**	0.7586±0.0213	**0.8871±0.0044**	0.7270±0.0905	**0.7765±0.0508**	**0.6459±0.0691**
+NBA+LapPE	0.8650±0.0120	**0.7621±0.0172**	0.8870±0.0037	**0.7486±0.0612**	0.7647±0.0531	0.6378±0.0544
GAT	0.8694±0.0119	0.7463±0.0159	0.8787±0.0046	0.6054±0.0386	0.6000±0.0491	0.4757±0.0614
+NBA	**0.8722±0.0120**	0.7549±0.0171	**0.8829±0.0043**	0.6622±0.0514	0.7059±0.0562	**0.5838±0.0558**
+NBA+LapPE	0.8692±0.0098	**0.7561±0.0175**	0.8822±0.0047	**0.6730±0.0348**	**0.7314±0.0531**	0.5784±0.0640
GatedGCN	0.8477±0.0156	0.7325±0.0192	**0.8671** ±0.0060	0.6108±0.0652	0.5824±0.0641	0.5216±0.0987
+NBA	**0.8523±0.0095**	**0.7405±0.0187**	0.8661±0.0035	0.6162±0.0490	0.6431±0.0356	**0.5649±0.0532**
+NBA+LapPE	0.8517±0.0130	0.7379±0.0193	0.8661±0.0047	**0.6243±0.0467**	**0.6569±0.0310**	0.5405±0.0785

Shows improvements almost regardless of the backbone GNN