pathGCN: Learning General Graph Spatial Operators from Paths

Moshe Eliasof[*], Eldad Haber[†] and Eran Treister[*]

*Ben-Gurion University of the Negev

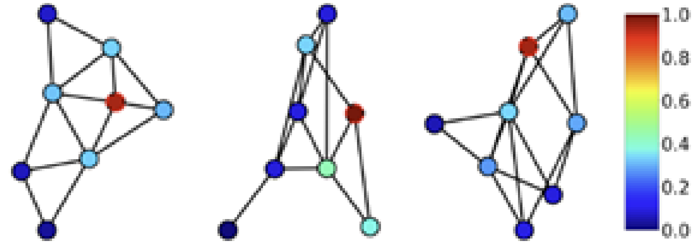
† University of British Columbia

ICML 2022

Popular types of GNNs

- GCN (Kipf & Welling, 2016) and subsequent works – utilize a proxy of the graph Laplacian as a spatial operator, followed by a 1×1 convolution.

$$\widetilde{\mathbf{P}} = \mathbf{I} - \tilde{\mathbf{L}} \qquad \mathbf{f}^{(l+1)} = \sigma\left(\widetilde{\mathbf{P}}\mathbf{f}^{(l)}\mathbf{W}^{(l)} \right)$$



- GAT (Veličković et al., 2018) – learns a non-negative weighting function of the edges

$$\alpha_{i,j}^{(l)} = \frac{\exp(LeakyReLU(\boldsymbol{a}^{(l)^{\boldsymbol{T}}}[\boldsymbol{W}^{(l)}\mathbf{f_i}||\boldsymbol{W}^{(l)}\mathbf{f_j}])}{\sum_{p \in \mathcal{N}_i} \exp(LeakyReLU(\boldsymbol{a}^{(l)^{\boldsymbol{T}}}[\boldsymbol{W}^{(l)}\mathbf{f_i}||\boldsymbol{W}^{(l)}\mathbf{f_p}])}$$

This weighting induces a spatial operator $\widetilde{\boldsymbol{A}}_{i,j}^{(l)} = \alpha_{i,j}^{(l)}$ that is applied to the graph nodes.

Why do we need to learn spatial operators?

- In CNNs, spatial operators are general (i.e., mixed signs) and can be:
  - Shared across channels
  - Per-channel (i.e., depth-wise convolutions)
  - Mixed with channel-wise convolutions (e.g., fully connected convolutions)
- In GNNs, however, typically the scope of spatial operators is limited:
  - GCNs and GATs employ non-negative operators shared across the channels.
  - The spatial operator is limited to pairwise interactions (kernel of size 2).
- This is a gap that we aim to bridge using Graph Random Walks:
  - Allowing to learn a wide spatial operator **per-layer and per-channel**.
  - Learning a **general** (i.e., with mixed signs) convolution operator.

From fixed to variable spatial operators

- Given a path $y_j = (j_0, \dots, j_{k-1})$, the spatial operator of size k is parameterized by a vector $\mathbf{s} \in \mathbb{R}^k$ and defined as follows:

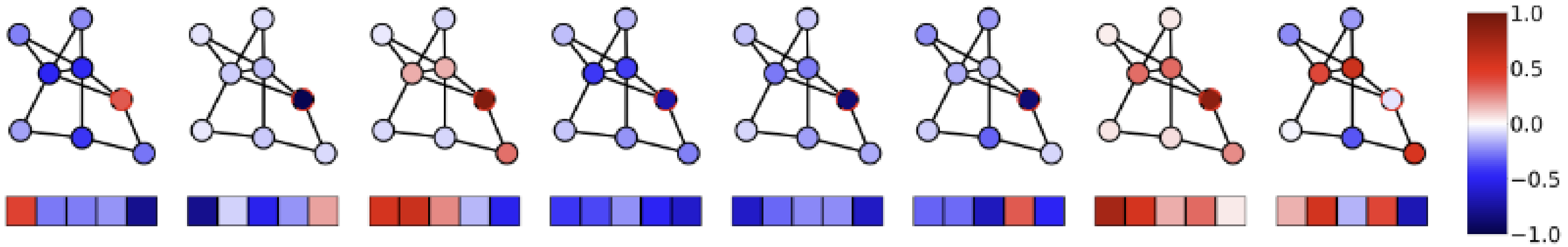
$$\mathbf{K}_{y_j}(\mathbf{s})\mathbf{f} = \sum_{i=0}^{k-1} \mathbf{s}_i \mathbf{f}_{j_i}$$

- In practice, we will sample multiple paths denoted by $\mathcal{Y}_j$ and average their responses:

$$\mathbf{K}_{\mathcal{Y}_j}(\mathbf{s})\mathbf{f} = \sum_{y_j \in \mathcal{Y}_j} \mathbf{K}_{y_j}(\mathbf{s})\mathbf{f}$$

$\mathbf{K}_{\mathcal{Y}_j}(\mathbf{s})\mathbf{f}$

$\boldsymbol{s}$



Constructing pathGCN

- Given the path convolution operator $\mathbf{K}_y$ we define a pathGCN layer as:

$$\mathbf{f}^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{K}_y^{(l)}\mathbf{f}^{(l)})$$

  where σ is an activation function and $\mathbf{W}^{(l)}$ is a 1×1 convolution operator.

- We can define three different types of pathGCN:
  - Global pathGCN – parameterized by a shared, global vector $s \in \mathbb{R}^k$.
  - Per-layer pathGCN – learning a spatial filter per layer $s \in \mathbb{R}^{L \times k}$.
  - Depth-wise pathGCN – a spatial operator per layer and channel $s \in \mathbb{R}^{L \times c \times k}$.

- Those variants are more similar to the ones found in CNNs (c.f., MobileNets), and offer a rich bank of spatial operators, compared to using the standard $\widetilde{\boldsymbol{P}}$ from GCNs.

Experiments

Semi-supervised node classification

- Our pathGCN improves with more layers and does not over-smooth.
- Both deterministic and stochastic implementations can be used for inference.

Inference	Cora	Cite.	Pub.
Determin.	85.8	75.8	82.7
Stochastic	85.8 $\pm$0.29	75.8 $\pm$0.34	82.7 $\pm$0.34

Dataset	Method	Layers 2	4	8	16	32	64
Cora	GCN	**81.1**	80.4	69.5	64.9	60.3	28.7
	GCN (Drop)	**82.8**	82.0	75.8	75.7	62.5	49.5
	JKNet	–	80.2	80.7	80.2	**81.1**	71.5
	JKNet (Drop)	–	**83.3**	82.6	83.0	82.5	83.2
	Incep	–	77.6	76.5	81.7	**81.7**	80.0
	Incep (Drop)	–	82.9	82.5	83.1	83.1	**83.5**
	GCNII	82.2	82.6	84.2	84.6	85.4	**85.5**
	GCNII*	80.2	82.3	82.8	83.5	84.9	**85.3**
	PDE-GCN$_D$	82.0	83.6	84.0	84.2	84.3	**84.3**
	EGNN	83.2	–	–	85.4	–	**85.7**
	pathGCN (Ours)	84.2	84.5	84.6	85.1	85.4	**85.8**
Citeseer	GCN	**70.8**	67.6	30.2	18.3	25.0	20.0
	GCN (Drop)	**72.3**	70.6	61.4	57.2	41.6	34.4
	JKNet	–	68.7	67.7	**69.8**	68.2	63.4
	JKNet (Drop)	–	72.6	71.8	**72.6**	70.8	72.2
	Incep	–	69.3	68.4	**70.2**	68.0	67.5
	Incep (Drop)	–	**72.7**	71.4	72.5	72.6	71.0
	GCNII	68.2	68.8	70.6	72.9	**73.4**	73.4
	GCNII*	66.1	66.7	70.6	72.0	**73.2**	73.1
	PDE-GCN$_D$	74.6	75.0	75.2	75.5	**75.6**	75.5
	pathGCN (Ours)	74.3	74.8	75.4	75.3	75.6	**75.8**
Pubmed	GCN	**79.0**	76.5	61.2	40.9	22.4	35.3
	GCN (Drop)	**79.6**	79.4	78.1	78.5	77.0	61.5
	JKNet	–	78.0	**78.1**	72.6	72.4	74.5
	JKNet (Drop)	–	78.7	78.7	**79.7**	79.2	78.9
	Incep	–	77.7	**77.9**	74.9	–	–
	Incep (Drop)	–	**79.5**	78.6	79.0	–	–
	GCNII	78.2	78.8	79.3	**80.2**	79.8	79.7
	GCNII*	77.7	78.2	78.8	**80.3**	79.8	80.1
	PDE-GCN$_D$	79.3	**80.6**	80.1	80.4	80.2	80.3
	EGNN	79.2	–	–	80.0	–	**80.1**
	pathGCN (Ours)	81.8	81.8	82.4	82.5	82.4	**82.7**

Fully-supervised node classification

Method	Cora	Cite.	Pubm.	Cham.	Corn.	Texas	Wisc.
GCN (Kipf & Welling, 2016)	85.77	73.68	88.13	28.18	52.70	52.16	45.88
GAT (Veličković et al., 2018)	86.37	74.32	87.62	42.93	54.32	58.38	49.41
Geom-GCN-I (Pei et al., 2020)	85.19	77.99	90.05	60.31	56.76	57.58	58.24
Geom-GCN-P (Pei et al., 2020)	84.93	75.14	88.09	60.90	60.81	67.57	64.12
Geom-GCN-S (Pei et al., 2020)	85.27	74.71	84.75	59.96	55.68	59.73	56.67
APPNP (Klicpera et al., 2019)	87.87	76.53	89.40	54.30	73.51	65.41	69.02
JKNet (Xu et al., 2018)	85.25 (16)	75.85 (8)	88.94 (64)	60.07 (32)	57.30 (4)	56.49 (32)	48.82 (8)
JKNet (Drop) (Rong et al., 2020)	87.46 (16)	75.96 (8)	89.45 (64)	62.08 (32)	61.08 (4)	57.30 (32)	50.59 (8)
Incep (Drop) (Rong et al., 2020)	86.86 (8)	76.83 (8)	89.18 (4)	61.71 (8)	61.62 (16)	57.84 (8)	50.20 (8)
GCNII (Chen et al., 2020)	88.49 (64)	77.08 (64)	89.57 (64)	60.61 (8)	74.86 (16)	69.46 (32)	74.12 (16)
GCNII* (Chen et al., 2020)	88.01 (64)	77.13 (64)	90.30 (64)	62.48 (8)	76.49 (16)	77.84 (32)	81.57 (16)
PDE-GCN$_M$(Eliasof et al., 2021)	88.60 (16)	78.48 (32)	89.93 (16)	66.01 (16)	89.73 (64)	93.24 (32)	91.76 (16)
pathGCN (Ours)	**90.02** (64)	**78.95** (32)	**90.42** (64)	**66.79** (16)	**91.35** (8)	**95.14** (16)	**93.53** (16)

Graph classification

- Our pathGCN is also useful for graph classification tasks.

Model	MUTAG	PTC	PROTEINS	NCI1
DGCNN	$85.8_{\pm 1.8}$	$58.6_{\pm 2.5}$	$75.5_{\pm 0.9}$	$74.4_{\pm 0.5}$
IGN	$83.9_{\pm 13.0}$	$58.5_{\pm 6.9}$	$76.6_{\pm 5.5}$	$74.3_{\pm 2.7}$
GIN	$89.4_{\pm 5.6}$	$64.6_{\pm 7.0}$	$76.2_{\pm 2.8}$	$82.7_{\pm 1.7}$
CIN	$92.7_{\pm 6.1}$	$68.2_{\pm 5.6}$	$77.0_{\pm 4.3}$	$\mathbf{83.6}_{\pm \mathbf{1.4}}$
GSN	$92.7_{\pm 7.5}$	$68.2_{\pm 7.2}$	$76.6_{\pm 5.0}$	$83.5_{\pm 2.0}$
pathGCN (ours)	$\mathbf{94.7}_{\pm \mathbf{4.7}}$	$\mathbf{75.2}_{\pm \mathbf{5.3}}$	$\mathbf{80.4}_{\pm \mathbf{4.2}}$	$83.3_{\pm 1.3}$

Summary

- A new approach for learning the spatial operators of GNNs is proposed
  - Based on graph random walks
  - Obtain a general spatial operator (i.e., mixed signs operator) with a larger aperture.
- An extensive set of experiments is carried to verify the method, achieving new SOTA performance:
  - Semi & Fully supervised node classification
  - Inductive Learning on PPI
  - Graph classification
  - Ablation study – variants of pathGCN & influence of kernel size and sampling density (see main paper)
- CNNs employ rich spatial operators with mixed signs and typically do not over-smooth – we bridge this gap between CNNs and GNNs.

THANK YOU FOR LISTENING
☺