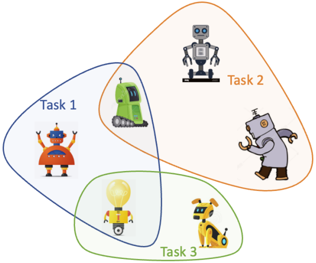
T-HyperGNNs: Hypergraph Neural Networks Via Tensor Representations

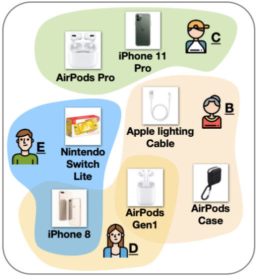
Graph Signal Processing Workshop 2023

Fuli Wang, Karelia Pena-Pena, Wei Qian, Gonzalo R.Arce

University of Delaware, USA

June 12, 2023

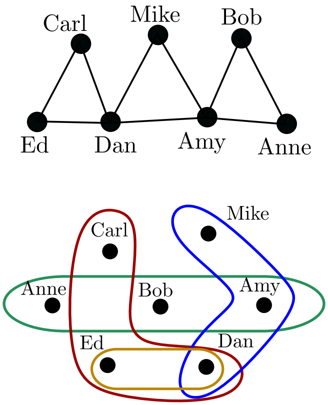




1. Introduction
2. Hypergraph Signal Shifting
3. T-Message Passing
4. Experiment

Hypergraph

Graph VS Hypergraph



Co-authorship Network

A hypergraph $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$

$V(\mathbf{H}) = \{\mathbf{v}_1, ...\mathbf{v}_N\}$ nodes •

$E(\mathbf{H}) = \{\mathbf{e}_1, ...\mathbf{e}_E\}$ hyperedges

$M = m.c.e(\mathbf{H}) = \max\{|\mathbf{e}_i| : \mathbf{e}_i \in E(\mathbf{H})\}$

maximum cardinality of the hyperedges

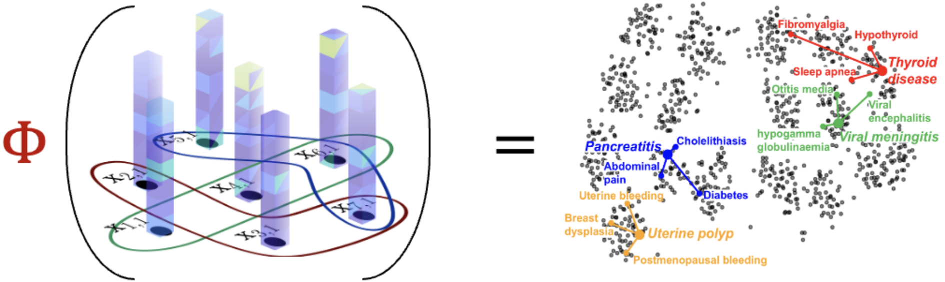
In the hypergraph example,

$|V(\mathbf{H})| = 7, \quad |E(\mathbf{H})| = 4, \quad M = 3$

A graph $\mathbf{G} = (V(\mathbf{G}), E(\mathbf{G}))$ is a uniform hypergraph where all edges have size 2 $(M = 2)$.

Hypergraph Neural Networks (HyperGNNs)

Goal: Learn the representation function $\mathbf{\Phi} : (\mathbf{H}, \mathbf{X}) \to \mathbb{R}$, where $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$ is the hypergraph structure and $\mathbf{X}$ is the signal



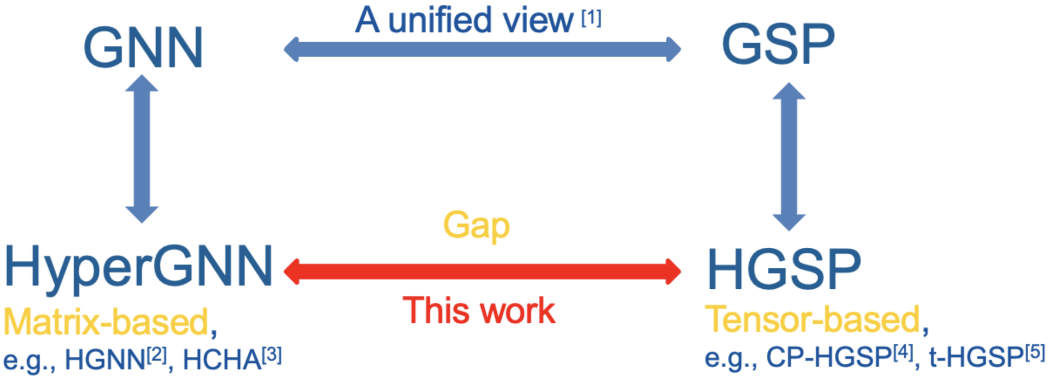
Input hypergraph + signal

Latent representation space

- Hypergraph signal shifting: $\mathbf{Y} = \phi_{shift}(\mathbf{X}, \mathbf{H})$
- hypergraph signal transformation: $\mathbf{Z} = \mathrm{MLP}_{\mathcal{W}}(\mathbf{Y})$

Motivation of This Work

Hypergraph signal shifting is a well-defined operation in HGSP using tensors.



[1] Gama, F., et al. IEEE Signal Processing Magazine (2020)
[2] Feng, Y., et al., AAAI (2019)
[3] Song, B., et al., Pattern Recognition (2021)
[4] Zhang, S., et al., IEEE Internet of Things Journal (2019)
[5] Pena, K., et al., IEEE Transactions on Signal and Information Processing over Networks (2023)

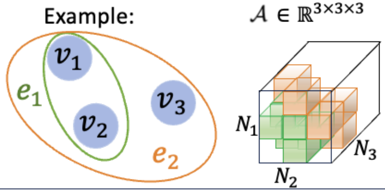
Hypergraph Adjacency Tensor

A hypergraph $\mathbf{H} = (V(\mathbf{H}), E(\mathbf{H}))$ can be represented by an Mth-order N-dimensional **Adjacency tensor** $\mathcal{A} \in \mathbb{R}^{N^M}$.

$$a_{n_1,n_2,\dots,n_M} = \begin{cases} \frac{|\mathbf{e}|}{\alpha}, & \text{if } \mathbf{e} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c\} \in E(\mathbf{H}) \text{ forms a hyperedge} \\ 0 & \text{otherwise,} \end{cases}$$

where α is the total number of permutations for length-M edge e^M.

Example:

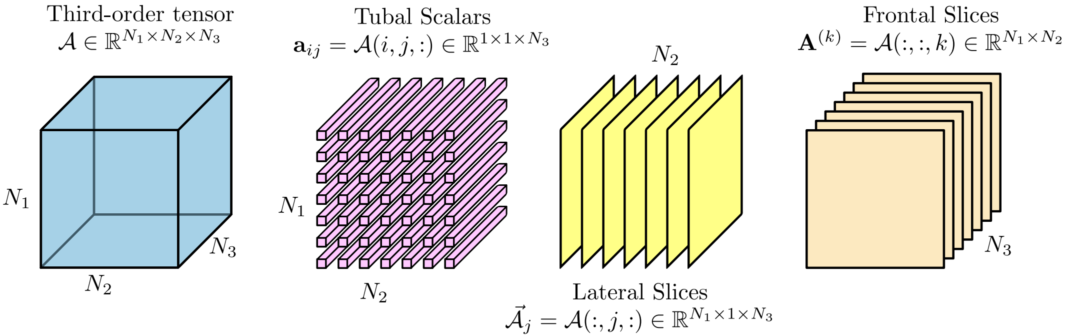


e_1: $a_{121} = a_{112} = a_{122} = a_{211} = a_{212} = a_{221} = 2/6 = 1/3$

e_2: $a_{123} = a_{132} = a_{213} = a_{231} = a_{312} = a_{321} = 3/6 = 1/2$

Zhang, S., et al. "Introducing hypergraph signal processing." IEEE Internet of Things Journal (2019).

Notation


Third-order tensor
$\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$
N_1
N_2
N_3
Tubal Scalars
$\mathbf{a}_{ij} = \mathcal{A}(i, j, :) \in \mathbb{R}^{1 \times 1 \times N_3}$
N_1
N_2
N_2
Lateral Slices
$\vec{\mathcal{A}}_j = \mathcal{A}(:, j, :) \in \mathbb{R}^{N_1 \times 1 \times N_3}$
Frontal Slices
$\mathbf{A}^{(k)} = \mathcal{A}(:, :, k) \in \mathbb{R}^{N_1 \times N_2}$
N_3

Hypergraph Signals Modeling Interaction

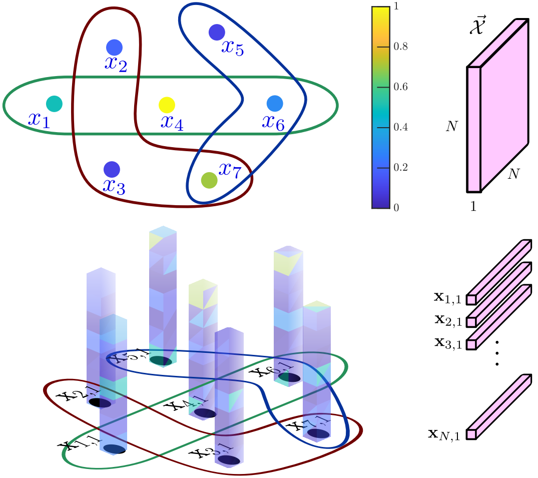
- A hypergraph signal is an N-length vector of tubal scalars $\vec{\mathcal{X}} \in \mathbb{R}^{N\times 1\times N}$.
- Each tubal scalar $\mathbf{x}_{i,1}$ in $\vec{\mathcal{X}}$ is obtained from a one dimensional signal in the hypergraph $\mathbf{x} \in \mathbb{R}^N$ as

$$\mathbf{x}_{i,1} = \texttt{fold}\left(\begin{bmatrix} x_i x_1 \\ x_i x_2 \\ \vdots \\ x_i x_N \end{bmatrix}\right).$$

- D-dimensional hypergraph signal:

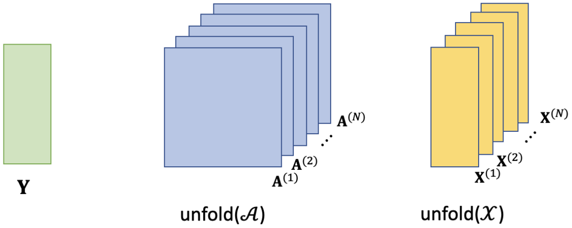
$$\mathcal{X} = \texttt{stack}([\vec{\mathcal{X}}_1, ..., \vec{\mathcal{X}}_D,], dim = 2)$$

$$\mathcal{X} \in \mathbb{R}^{N\times D\times N}$$



Zhang, S., et al. "Introducing hypergraph signal processing." IEEE Internet of Things Journal (2019).

Hypergraph Signal Shifting



$\mathcal{A} \in \mathbb{R}^{N \times N \times N}$
$\mathcal{X} \in \mathbb{R}^{N \times D \times N}$
$\mathbf{Y} \in \mathbb{R}^{N \times D}$

Time complexity: $\mathcal{O}(N^M)$
Space complexity: $\mathcal{O}(N^M)$

$$\mathbf{Y} := \mathcal{A} \cdot \mathcal{X} = \texttt{unfold}(\mathcal{A}) \cdot \texttt{unfold}(\mathcal{X})$$

$$= \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(2)} & \cdots & \mathbf{A}^{(N-1)} & \mathbf{A}^{(N)} \end{bmatrix} \begin{bmatrix} \mathbf{X}^{(1)} \\ \mathbf{X}^{(2)} \\ \vdots \\ \mathbf{X}^{(N-1)} \\ \mathbf{X}^{(N)} \end{bmatrix} = \sum_{k=1}^{N} \mathbf{A}^{(k)} \mathbf{X}^{(k)}$$

To do: Scale up the hypergraph signal shifting operation.

Geometric Meaning of the Hypergraph Signal Shifting

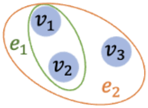
Hypergraph perspective:

$$\mathbf{Y} := \mathcal{A} \cdot \mathcal{X} = \sum_{k=1}^{N} \mathbf{A}^{(k)} \mathbf{X}^{(k)}$$

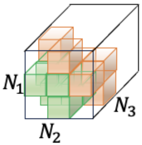
Node perspective:

$$[\mathbf{Y}]_{i,d} = \sum_{j=1}^{N} \sum_{k=1}^{N} a_{ijk} x_{j_d} x_{k_d},$$

Example:

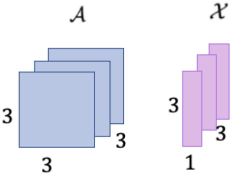


$\mathcal{A} \in \mathbb{R}^{3\times3\times3}$



e_1: $a_{121} = a_{112} = a_{122} = a_{211} = a_{212} = a_{221} = 2/6 = 1/3$

e_2: $a_{123} = a_{132} = a_{213} = a_{231} = a_{312} = a_{321} = 3/6 = 1/2$



$$\mathbf{Y} = \begin{bmatrix} \frac{2}{3}x_1x_2 + \frac{1}{3}x_2^2 + x_2x_3 \\ \frac{2}{3}x_1x_2 + \frac{1}{3}x_1^2 + x_1x_3 \\ x_1x_2 \end{bmatrix}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{N} a_{ijk} x_j x_k$$

$a_{121}x_2x_1 + a_{112}x_1x_2 + a_{122}x_2x_2 + a_{123}x_2x_3 + a_{132}x_3x_2$

$a_{221}x_2x_1 + a_{212}x_1x_2 + a_{211}x_1x_1 + a_{213}x_1x_3 + a_{231}x_3x_1$

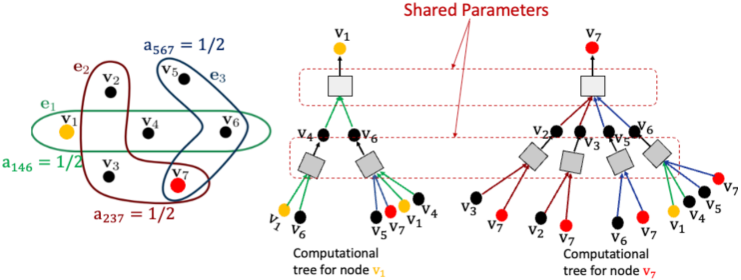
$a_{312}x_1x_2 + a_{321}x_2x_1$

Takeaway: Focusing on the connectivity of each node.

T-Message Passing HyperGNNs

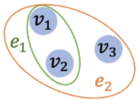
Idea: Neighboring nodes pass “message" to the central node

$$[\mathbf{Y}]_v = \text{AGGREGATE}(\{\mathbf{x}_u, \forall u \in \mathcal{N}(v)\}),$$



T-Message Passing HyperGNNs

Compressed Adjacency Tensor: Adjacency value table + Neighborhood table



(a) A hypergraph

e_1: $a_{121} = a_{112} = a_{122} = a_{211} = a_{212} = a_{221} = 2/6 = 1/3$

e_2: $a_{123} = a_{132} = a_{213} = a_{231} = a_{312} = a_{321} = 3/6 = 1/2$

(b) Nonzero adjacency tensor entries

⇓ Compress

|e|	a_e
2	1/3
3	1/2

(c) The adjacency value table

v	$\mathcal{N}^3(v)$
v_1	$\{\{\pi(v_2,v_1),\pi(v_2,v_2)\},\{\pi(v_2,v_3)\}\}$
v_2	$\{\{\pi(v_1,v_2),\pi(v_1,v_1)\},\{\pi(v_1,v_3)\}\}$
v_3	$\{\pi(v_1,v_2)\}$

(d) The neighborhood table

Space complexity: Reduced from $\mathcal{O}(N^M)$ to $\mathcal{O}(N)$

Define the M-order incidence edge set and M-order neighborhood of v as

$$E^M(v) = \{e^M | e \ni v\},$$
$$\mathcal{N}^M(v) = \{\pi(e^M(-v)) | \forall e^M \in E^M(v)\},$$

where $e^M(-v)$ refers to deleting exact one v element from e^M, and $\pi(\cdot)$ represents permutation of a sequence. E.g., For node v_1,

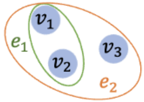
$$E^M(v_1) = \{\underbrace{[(v_1,v_2,v_1),(v_1,v_2,v_2)]}_{e_1^3}, \underbrace{(v_1,v_2,v_3)}_{e_2^3}\},$$

$$\mathcal{N}^M(v_1) = \{\pi(v_2,v_1),\pi(v_2,v_2),\pi(v_2,v_3)\}$$

T-Message Passing HyperGNNs

Aggregate with **Compressed Adjacency Tensor**:

Example:

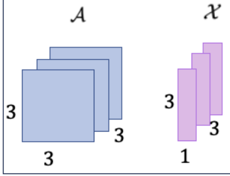


$\lvert e\rvert$	a_e
2	1/3
3	1/2

The adjacency value table

v	$\mathcal{N}^3(v)$
v_1	$\{\{\pi(v_2,v_1),\pi(v_2,v_2)\},\{\pi(v_2,v_3)\}\}$
v_2	$\{\{\pi(v_1,v_2),\pi(v_1,v_1)\},\{\pi(v_1,v_3)\}\}$
v_3	$\{\pi(v_1,v_2)\}$

The neighborhood table



$$\mathbf{Y} = \begin{bmatrix} \frac{2}{3}x_1x_2 + \frac{1}{3}x_2^2 + x_2x_3 \\ \frac{2}{3}x_1x_2 + \frac{1}{3}x_1^2 + x_1x_3 \\ x_1x_2 \end{bmatrix}$$

$$[\mathbf{Y}]_v = \underbrace{\sum_{e^M \in E^M(v)}}_{\text{different edges}} \underbrace{\Big(a_e \sum_{\pi(\cdot)\in\mathcal{N}^M(v)} \prod_{u\in\pi(\cdot)} \mathbf{x}_u\Big)}_{\text{interaction in an } e}$$

E.g. $[\mathbf{Y}]_{v_1} = a_{e_1}(x_1x_2 + x_2x_1 + x_2^2) + a_{e_2}(x_2x_3 + x_3x_2) = \frac{2}{3}x_1x_2 + \frac{1}{3}x_2^2 + x_2x_3$

Time complexity: Reduced from $\mathcal{O}(N^M)$ to $\mathcal{O}(Nd_{max})$, where d_{max} is the maximum degree of nodes.

T-HyperGNNs

From hypergraph signal shifting

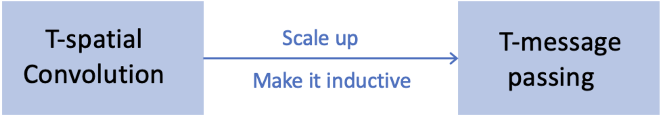
- Performing signal transformation with **learnable weights**.
- Cascading **multiple layers** to form T-HyperGNNs.

$$\begin{cases} \mathbf{X}^{(l+1)} = \sigma(\mathcal{A} \cdot \mathcal{X}^{(l)} \mathbf{W}^{(l)}), \\ \mathcal{X}^{(l)} = \texttt{HGS}(\mathbf{X}^{(l)}) \end{cases}$$

$$\begin{cases} \mathbf{x}_v^{(l)} \leftarrow \sigma(\mathsf{MLP}^{(l)}(\mathsf{COMBINE}(\mathbf{x}_v^{(l-1)}, [\mathbf{Y}]_v^{(l)}))) \\ [\mathbf{Y}]_v^{(l)} = \underset{e^M \in E^M(v)}{\mathsf{AGGREGATE}}(a_e \sum_{\pi(\cdot) \in \mathcal{N}^M(v)} \prod_{u \in \pi(\cdot)} (\mathbf{x}_u^{(l-1)})) \end{cases}$$

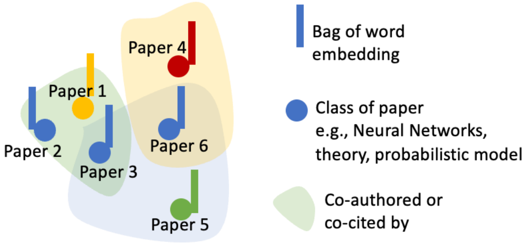
$$\mathbf{Y} = \mathcal{A} \cdot \mathcal{X}$$

$$[\mathbf{Y}]_v = \sum_{e^M \in E^M(v)} (a_e \sum_{\pi(\cdot) \in \mathcal{N}^M(v)} \prod_{u \in \pi(\cdot)} \mathbf{x}_u)$$



Experiment: Semi-supervised Node Classification

- Co-citation hypergraphs[1]:
  Cora, Citeseer, Pubmed
- Co-authorship hypergraphs[1]:
  Cora, DBLP



[1] Yadati, N., et al. "Hypergcn: A new method for training graph convolutional networks on hypergraphs." (2019)

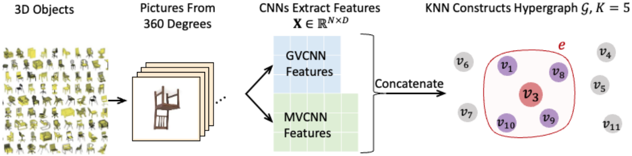
Experiment: Semi-supervised Node Classification

Method	Time Complexity⋆	Cocitation Cora	Cocitation Citeseer	Cocitation Pubmed	Coauthorship Cora	Coauthorship DBLP
MLP	$\mathcal{O}(ND^2)$	48.23 ± 7.35	65.56 ± 1.48	73.89 ± 5.60	46.11 ± 8.35	76.15 ± 7.26
HGNN[1]	$\mathcal{O}(ND^2 + N^2D)$	70.59 ± 1.22	73.89 ± 8.98	82.22 ± 1.33	66.94 ± 6.51	93.08 ± 6.39
HyperGCN[2]	$\mathcal{O}(ND^2 + N^2D + E\delta_e)$	35.29 ± 1.24	61.11 ± 1.53	76.11 ± 1.40	25.79 ± 6.43	25.38 ± 1.29
HNHN[3]	$\mathcal{O}(ND^2 + NED + ED^2)$	69.41 ± 9.04	74.44 ± 9.69	77.22 ± 4.08	71.39 ± 5.56	93.85 ± 5.76
T-spatial	$\mathcal{O}(N^M D + N^{(M-1)} D^2)$	69.17 ± 7.58	76.11 ± 7.05	84.22 ± 3.26	70.00 ± 6.01	94.62 ± 2.93
T-MPHN	$\mathcal{O}(ND^2 + N\delta_v)$	**70.83 ± 5.59**	**77.22 ± 6.44**	**93.33 ± 4.48**	**72.78 ± 4.44**	**95.38 ± 2.10**

[1] Feng, Y., et all. AAAI. (2019) [2] Yadati, N., et all. NIPS. (2019) [3] Dong, Y., et all. arXiv. (2020)

⋆ Time complexity for one layer of HyperGNNs. δ_v is the maximum node degree, δ_e is the maximum edge degree. N is the number of nodes, E is the number of edges, and D is the hidden dimension.

Experiment: Inductive 3D-object Detection



Statistic	ModelNet40	NTU
$\lvert\mathcal{V}\rvert$	12311	2012
$\lvert\mathcal{E}\rvert$	24622	4024
Feature Dimension D	6144	6144
Number of Classes	40	67

Method	Time Complexity⋆	ModelNet40[3]			NTU[4]		
		Seen	Unseen	Reduced (%)	Seen	Unseen	Reduced (%)
MLP	$\mathcal{O}(ND^2)$	96.13 ± 2.17	88.42 ± 1.41	8.72	94.51 ± 4.70	77.68 ± 4.46	17.81
HyperSAGE[1]	$\mathcal{O}(ND^2 + E\delta_e)$	97.55 ± 2.35	88.37 ± 2.66	10.39	97.33 ± 3.58	75.34 ± 1.04	22.59
UniSAGE[2]	$\mathcal{O}(ND^2 + N\delta_v)$	$\mathbf{100.00 \pm 0.00}$	92.62 ± 2.19	7.38	96.60 ± 1.43	81.05 ± 0.82	16.10
T-MPHN	$\mathcal{O}(ND^2 + N\delta_v)$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{96.69 \pm 3.22}$	$\mathbf{3.31}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{86.34 \pm 2.17}$	$\mathbf{13.66}$

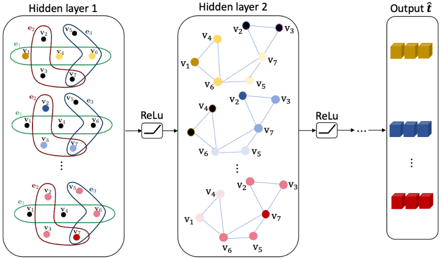
[1] Arya, D., et al. arXiv (2020) [2] Huang, J., et al. ijcai. (2021). [3] Wu, Z., et al. CVPR (2015) [4] Chen, D., et al. Wiley (2003).

⋆ Time complexity for one layer of HyperGNNs. δ_v is the maximum node degree, δ_e is the maximum edge degree. N is the number of nodes, E is the number of edges, and D is the hidden dimension.

Variation of T-MPHN

Variation on Aggregation:

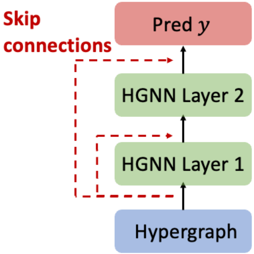
- Change the order of hypergraph M at different layers



- Change the aggregate function, e.g, Mean, attention weighted sum, etc

Variation on Combining:

- COMBINE = Concatenation
  $\mathbf{x}_v^{(l)} \leftarrow \sigma(\text{MLP}^{(l)}([\mathbf{x}_v^{(l-1)}; [\mathbf{Y}]_v^{(l)}]))$
- COMBINE with skip-connection
  $\mathbf{x}_v^{(l)} = \sigma(\text{MLP}^{(l)}([\mathbf{x}_v^{(0)}; \mathbf{x}_v^{(l-1)}; [\mathbf{Y}]_v^{(l)}]))$.



Acknowledgement

This work was supported by the NSF under grants 1815992, 1816003, and 1916376, the AFOSR award FA9550-22-1-0362 and by the Institute of Financial Services Analytics, sponsored by JP Morgan Chase & Co.

The manuscript is submitted to *IEEE Transactions on Neural Networks and Learning Systems* and the pre-print is available on TechRxiv, T-HyperGNNs: Hypergraph Neural Networks Via Tensor Representations.

Thank you!

Any Questions?

Extra Slides - Why Not Matrix?

A hypergraph can be represented by the incidence matrix $\mathbf{B} \in \mathbb{R}^{|V(\mathbf{H})| \times |E(\mathbf{H})|}$.

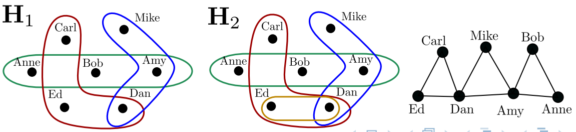
For the example hypergraph $\mathbf{H}_2$,

$$\mathbf{B} = \begin{array}{c} \\ \text{Anne} \\ \text{Carl} \\ \text{Ed} \\ \text{Bob} \\ \text{Mike} \\ \text{Amy} \\ \text{Dan} \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \end{array}, \quad \mathbf{A} = \begin{array}{c} \\ \text{Anne} \\ \text{Carl} \\ \text{Ed} \\ \text{Bob} \\ \text{Mike} \\ \text{Amy} \\ \text{Dan} \end{array} \begin{array}{c} \begin{array}{ccccccc} \text{Anne} & \text{Carl} & \text{Ed} & \text{Bob} & \text{Mike} & \text{Amy} & \text{Dan} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 1 & 2 & 0 & 1 & 1 & 3 \end{pmatrix} \end{array}$$

Projecting out the hyperedge dimension: Adjacency matrix $\mathbf{A} = \mathbf{B}\mathbf{B}^T$

$\Longrightarrow$ Clique expansion

not a one-to-one mapping



Extra Slides - Transductive VS Inductive

	Transductive	Inductive
Training	$(\mathbf{H}, \mathbf{X}) \to \mathbf{Z}_{train}$	$(\mathbf{H}_{train}, \mathbf{X}_{train}) \to \mathbf{Z}_{train}$
Testing	$(\mathbf{H}, \mathbf{X}) \to \mathbf{Z}_{test}$	$(\mathbf{H}_{test}, \mathbf{X}_{test}) \to \mathbf{Z}_{test}$

Extra Slides: Ablation Study for T-MPHN

Exam the effectiveness of

- Values of adjacency tensors
- Node interaction modeled by cross product