VilLain: Self-Supervised Learning on Homogeneous Hypergraphs without Features via Virtual Label Propagation

Geon Lee, Soo Yong Lee, and Kijung Shin
KAIST
{geonlee0325, syleetolow, kijungs}@kaist.ac.kr

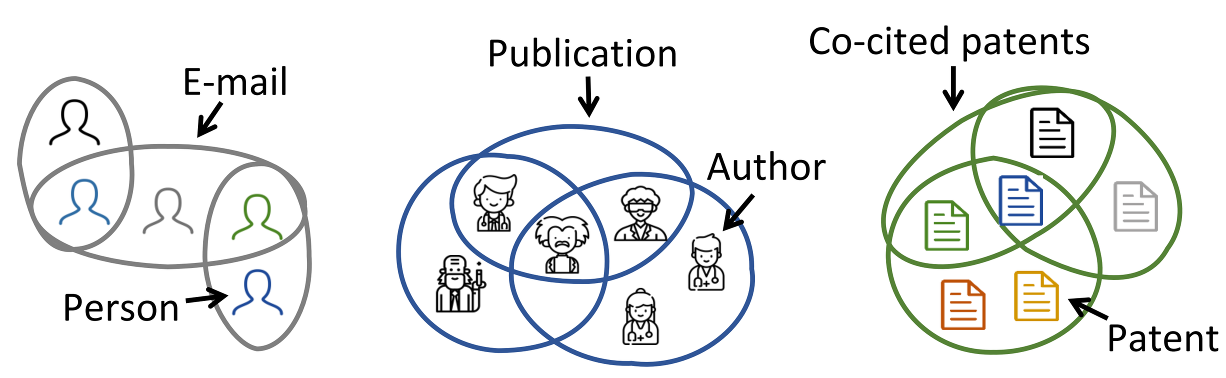
https://github.com/geon0325/VilLain

Summary

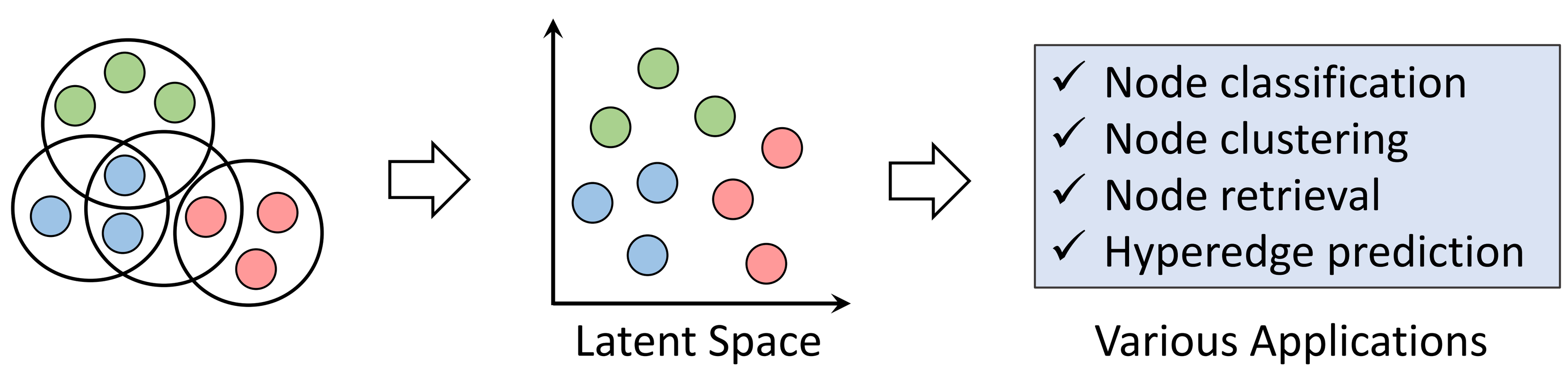
- **Goal:** to learn informative node embeddings in hypergraphs **without** any supervision (e.g., labels) or extra information (e.g., features)
- **Previous Work:**
  - Requires external node features.
  - Captures only structural information (no label information)
- **Proposed Method (VilLain):**
  - Inspired by **higher-order label homogeneity** in real hypergraphs;
  - Learns **virtual labels** (**v-labels**) to reproduce the observed pattern ***without relying on actual labels and input features***;
  - Generates embeddings from v-label probability assignment vectors → Captures **potential structure-label relationships**
- **Advantages:**
  - **Minimum Requirements:** Does not require any supervision (e.g., node labels) or external information (e.g., node features)
  - **Versatility:** Generates general-purpose embeddings
  - **Accuracy:** Outperforms the competitors up to 72% ↑ accuracy

Background: Hypergraphs

- **Hypergraphs** model **group interactions**.
  - Each **hyperedge** is a subset of any number of nodes

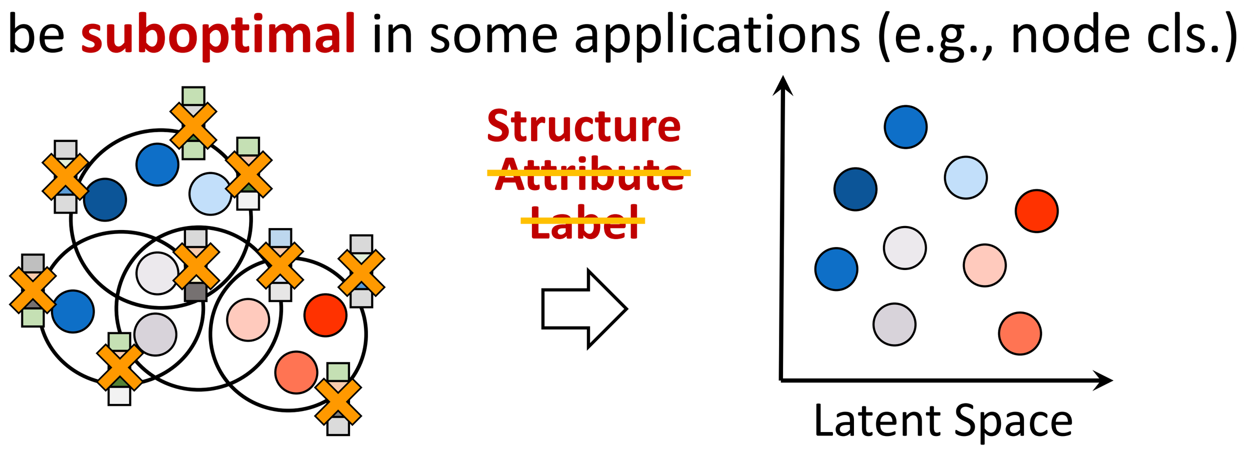


- **Hypergraph Representation Learning (HRL)** learns node embeddings.
  - **Self-supervised HRL** learns embeddings ***without external node labels***.



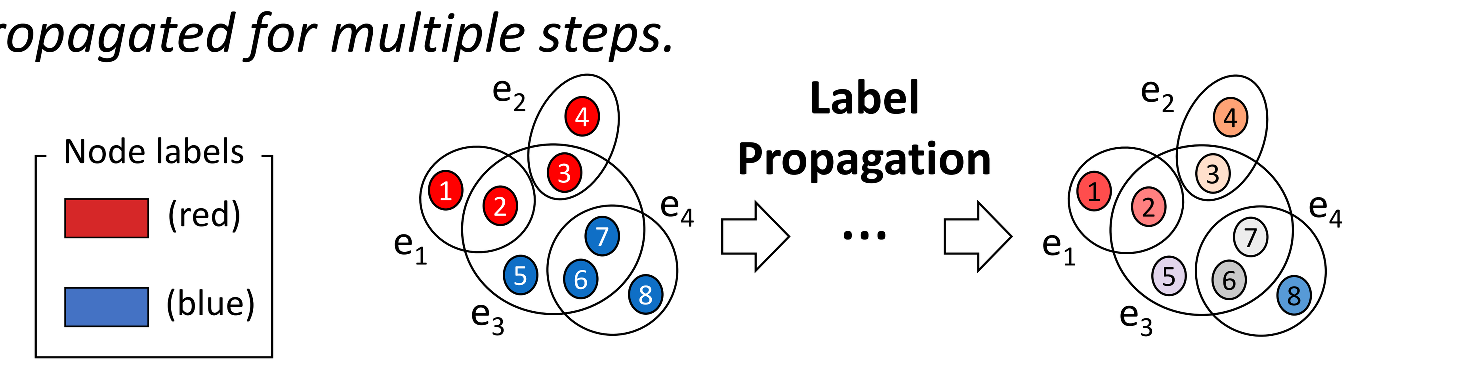
Motivations

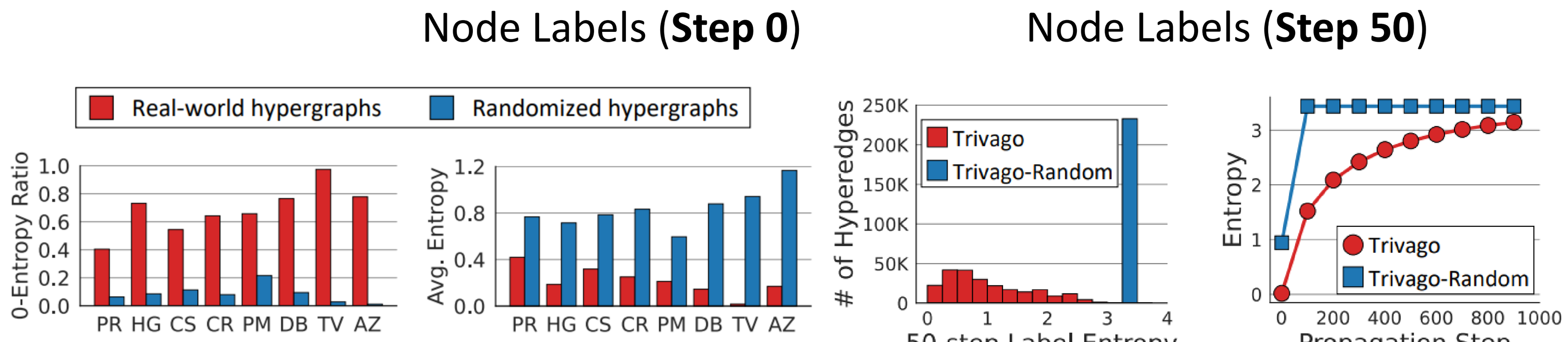
- **Most HRL methods require external features.**
  - External features are **often missing** in real-world hypergraphs.
- **Existing HRL methods that *do not require external features* consider only structural information.**
  - They do not consider **potential interplay with labels**.
  - They can be **suboptimal** in some applications (e.g., node cls.).



Observations

- **Obs 1. Label homogeneity in real-world hyperedges**
  - Hyperedges tend to contain the same labeled nodes.
- **Obs 2. Higher-order label homogeneity in real-world hyperedges**
  - Label homogeneity is maintained strong *even when labels are propagated for multiple steps.*



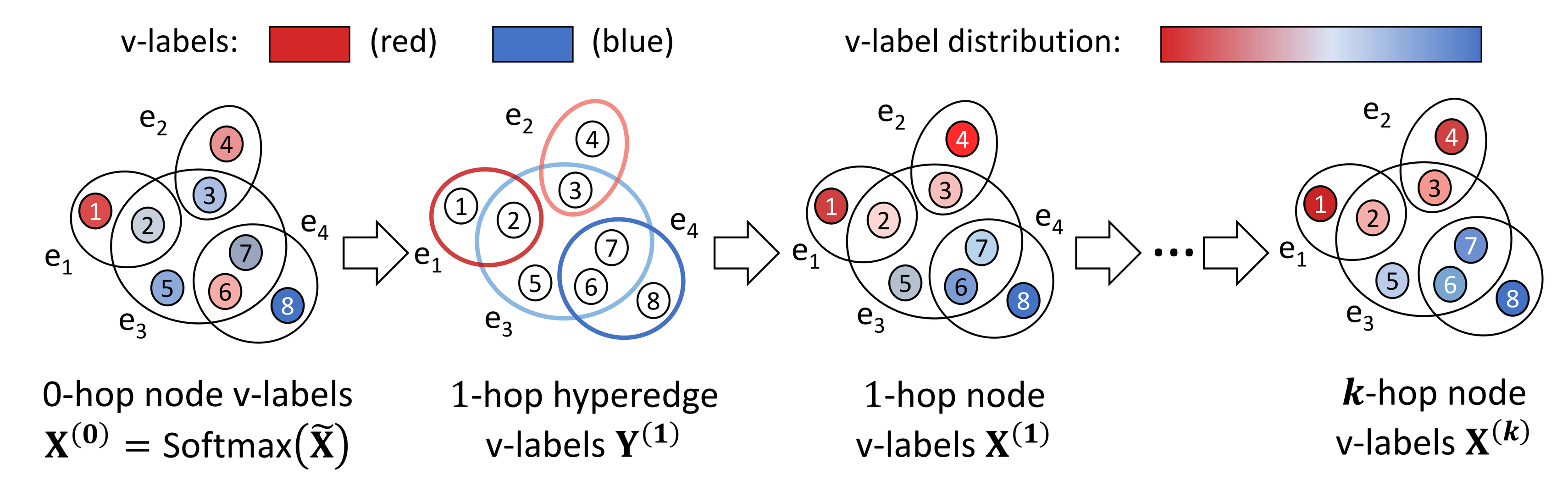


Observation 1

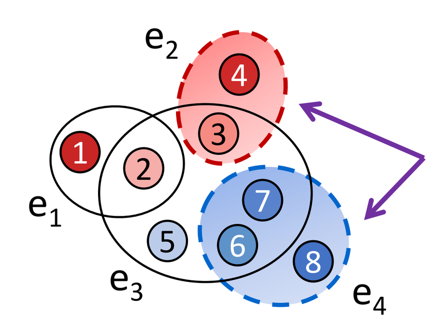
Observation 2

Proposed Method: VilLain

- **VilLain** learns versatile node embeddings for hypergraphs.
  - Learns distributions of **"virtual" labels** (**v-labels**) **w/o actual labels**;
  - V-labels are propagated across the hypergraph;
  - **Versatile node embeddings** are generated from v-label probability assignment vectors.

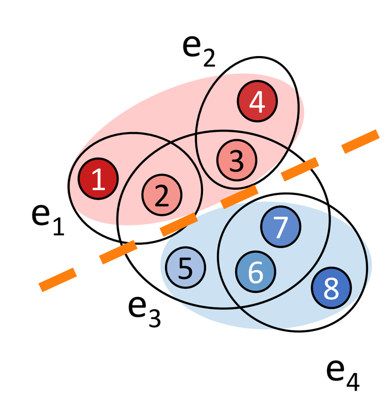


- **VilLain** aims to reproduce the **higher-order label homogeneity** present in real-world hypergraphs by optimizing v-label distributions.
  - Specifically, it aims to minimize the following **self-supervised losses**:



Local Loss

Capture the **higher-order homogeneity** of v-labels, motivated by the **observations** in real-world hypergraphs.



Make v-label distribution **distinctive** by inducing v-label clusters disjoint from each other.

Make v-label distribution **balanced** to prevent any single v-label's dominance.

Global Loss

Experimental Results

- **Q1. Accuracy: VilLain** outperforms graph and hypergraph

Method	DBLP	Trivago	Amazon	Primary	High	Citeseer	Cora	Pubmed	Rank
GCN	67.37 ± 1.45	38.06 ± 1.49	28.73 ± 4.73	75.63 ± 5.08	96.25 ± 2.55	60.64 ± 3.47	72.96 ± 1.82	77.56 ± 2.58	7.00 ± 2.91
GAT	61.74 ± 1.97	51.52 ± 0.68	30.94 ± 2.13	66.79 ± 4.73	90.58 ± 2.76	49.57 ± 2.64	58.09 ± 2.14	73.67 ± 1.78	11.75 ± 3.83
Deepwalk	29.03 ± 1.43	16.85 ± 0.45	25.43 ± 1.72	84.89 ± 3.67	99.31 ± 0.48	45.10 ± 3.18	56.58 ± 1.88	68.58 ± 2.60	11.62 ± 4.71
Node2vec	29.21 ± 1.89	16.88 ± 0.44	25.27 ± 2.36	83.53 ± 3.09	**99.38 ± 0.45**	45.37 ± 3.17	59.15 ± 1.84	69.05 ± 3.00	11.00 ± 4.35
DGI	62.37 ± 3.32	73.46 ± 1.22	31.80 ± 1.45	86.66 ± 4.51	92.49 ± 0.60	61.36 ± 2.91	71.23 ± 2.04	77.51 ± 1.38	7.25 ± 3.59
GRACE	71.86 ± 2.51	OOM	OOM	63.78 ± 5.12	99.03 ± 0.30	61.16 ± 2.78	73.43 ± 1.81	77.70 ± 1.81	5.50 ± 4.75
GMI	64.19 ± 1.63	OOM	OOM	80.10 ± 4.94	96.61 ± 2.63	58.67 ± 2.68	71.31 ± 1.69	75.51 ± 2.77	9.16 ± 1.57
HGNN	66.60 ± 2.18	OOM	OOM	88.28 ± 5.02	92.19 ± 3.84	60.91 ± 2.32	72.90 ± 2.00	76.58 ± 2.86	7.50 ± 3.09
HNHN	63.99 ± 2.21	59.52 ± 1.64	28.99 ± 2.63	91.31 ± 2.47	96.83 ± 1.25	59.02 ± 1.63	68.81 ± 1.26	75.33 ± 1.77	7.50 ± 2.39
AllSet	63.67 ± 1.89	36.58 ± 0.93	21.75 ± 1.67	85.94 ± 3.02	95.70 ± 1.66	56.08 ± 1.95	67.73 ± 1.81	74.11 ± 2.04	10.75 ± 1.08
UniGNN	67.16 ± 2.15	69.98 ± 1.60	33.77 ± 3.22	88.88 ± 3.58	95.12 ± 3.97	59.10 ± 2.76	71.44 ± 1.03	74.37 ± 2.10	7.12 ± 3.09
HyperGCL	58.72 ± 1.54	74.99 ± 1.23	22.86 ± 2.01	74.07 ± 6.06	85.79 ± 8.92	57.54 ± 1.61	74.99 ± 1.33	78.44 ± 3.33	8.87 ± 5.18
Hyper2vec	67.18 ± 1.78	75.82 ± 1.45	OOT	92.52 ± 2.45	96.34 ± 1.34	61.50 ± 2.60	71.79 ± 1.63	77.04 ± 1.51	4.85 ± 2.35
LBSN	22.63 ± 2.20	47.99 ± 0.82	11.56 ± 0.90	86.71 ± 3.71	95.87 ± 2.28	45.43 ± 2.15	59.70 ± 1.31	54.89 ± 2.38	11.87 ± 3.21
TriCL	68.18 ± 1.36	OOM	OOM	92.67 ± 2.50	98.10 ± 1.02	59.17 ± 3.35	72.35 ± 1.53	78.57 ± 1.88	4.16 ± 1.95
VilLain	**77.16 ± 1.26**	**79.43 ± 1.63**	**57.95 ± 2.47**	**93.66 ± 3.93**	99.19 ± 0.41	**61.53 ± 3.17**	**75.03 ± 1.38**	**78.82 ± 1.47**	**1.25 ± 0.66**

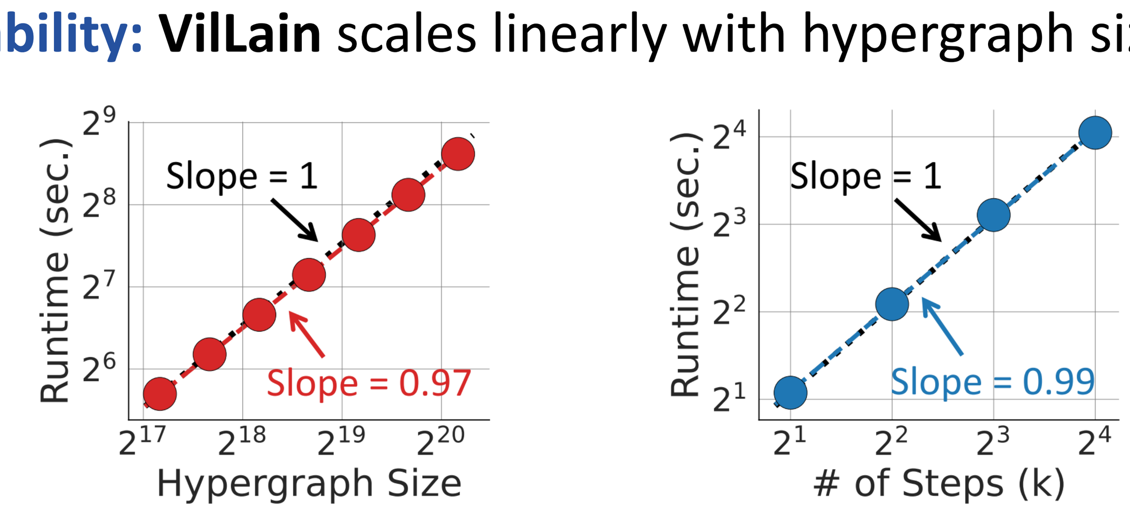
- **Q2. Versatility: VilLain** performs best in various downstream

Method	Hyperedge Prediction (Acc.) DB	TV	AZ	PR	HG	CS	CR	PM	Rank
Deepwalk	63.9	61.3	69.4	83.8	85.9	69.6	67.2	65.9	6.25
Node2vec	64.2	61.4	69.3	83.2	85.4	70.4	66.9	65.8	6.62
DGI	**86.1**	83.8	90.8	79.1	84.4	79.2	76.3	80.9	3.75
GRACE	85.4	OOM	OOM	80.3	87.4	77.9	74.5	79.1	4.00
GMI	75.6	OOM	OOM	82.4	85.9	74.4	69.4	72.3	5.50
Hyper2vec	71.2	72.4	OOT	76.4	79.6	78.1	71.7	71.5	6.14
LBSN	48.7	89.1	63.7	79.4	87.1	74.3	69.6	66.1	5.87
TriCL	77.4	OOM	OOM	**84.0**	87.8	82.0	76.7	80.5	2.33
VilLain	81.6	**95.1**	**94.9**	83.2	**87.8**	**82.1**	**79.0**	**82.8**	**1.50**

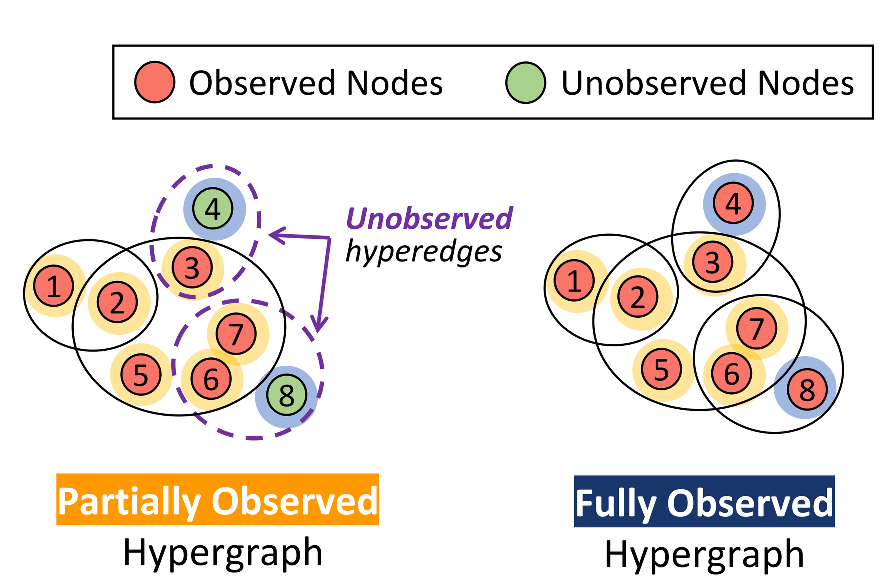
Method	Node Clustering (NMI) DB	TV	AZ	PR	HG	CS	CR	PM	Rank
Deepwalk	0.7	16.7	7.8	85.2	**100.0**	14.6	23.9	**34.4**	5.00
Node2vec	0.9	17.0	7.7	83.5	**100.0**	14.5	23.8	32.8	5.87
DGI	16.6	44.5	13.0	84.4	73.9	29.1	32.1	31.3	5.62
GRACE	43.0	OOM	OOM	67.6	98.2	33.0	46.0	31.6	5.00
GMI	27.8	OOM	OOM	84.1	93.1	25.3	42.6	18.7	6.50
Hyper2vec	43.4	66.3	OOT	**92.5**	99.3	34.3	45.5	33.6	2.27
LBSN	1.1	39.4	2.7	85.5	97.8	12.1	29.0	4.6	6.50
TriCL	38.0	OOM	OOM	87.8	98.7	34.4	44.8	33.7	3.00
VilLain	**46.6**	**69.4**	**35.2**	85.7	98.7	**34.5**	**50.4**	32.7	**2.25**

Method	Node Retrieval (MAP) DB	TV	AZ	PR	HG	CS	CR	PM	Rank
Deepwalk	21.3	7.5	27.7	81.6	98.7	27.6	29.2	49.0	6.37
Node2vec	21.6	7.1	27.8	81.1	98.6	27.3	29.4	49.4	6.50
DGI	36.1	37.3	31.1	89.7	97.8	43.8	50.6	61.7	3.25
GRACE	50.2	OOM	OOM	61.4	**99.5**	41.1	54.2	60.9	4.00
GMI	34.6	OOM	OOM	80.0	97.8	35.9	41.6	55.1	6.33
Hyper2vec	35.5	43.1	OOT	85.7	90.7	41.2	46.7	55.6	4.85
LBSN	21.0	19.1	29.1	81.3	93.2	30.6	40.1	43.5	6.62
TriCL	45.1	OOM	OOM	89.9	97.6	42.4	55.0	61.9	3.16
VilLain	**60.2**	**67.2**	**53.6**	**91.3**	99.0	**46.4**	**58.0**	**64.4**	**1.12**

- **Q3. Scalability: VilLain** scales linearly with hypergraph size and # steps.



- **Q4. Generalizability: VilLain** performs well on unobserved nodes.



Node Group 1 / Node Group 2

Model & Setting	Node Group	DB	TV	AZ	CS	CR	PM
VilLain (Fully Observed)	Group 1	78.01	80.03	56.77	62.75	75.43	79.21
	Group 2	66.19	76.07	58.74	57.52	73.46	69.62
VilLain (Partially Observed)	Group 1	76.45	78.66	53.72	62.49	73.79	77.78
	Group 2	65.86	74.71	55.23	56.42	72.27	69.71
TriCL	Group 1	69.20	OOM	OOM	60.83	72.94	78.99
	Group 2	55.11	OOM	OOM	53.74	70.02	68.60