An Introduction to Latent Graph Inference

Invited Talk at ACLab

Jianglin Lu

lu.jiang@northeastern.edu
https://jianglin954.github.io/

Feberary 26th, 2024

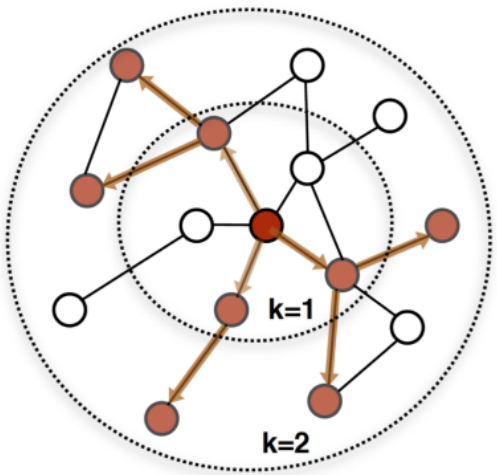
Outline

Outline

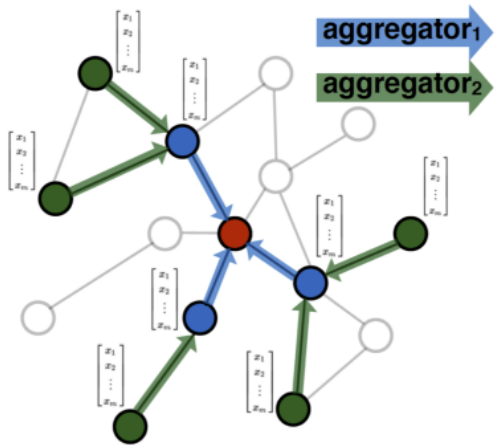
Outline

Graph Neural Networks

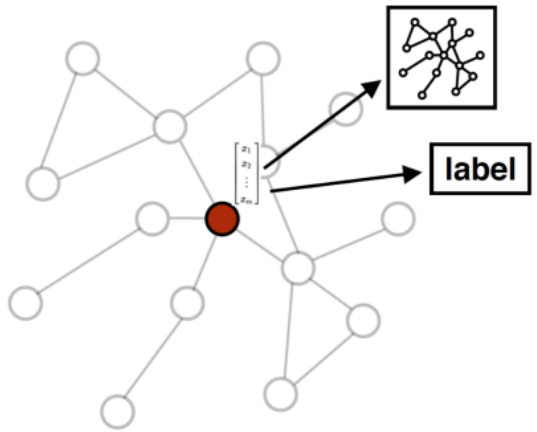
The key design element of GNNs is the use of pairwise message passing, such that graph nodes iteratively update their representations by exchanging information with their neighbors.



1. Sample neighborhood



2. Aggregate feature information from neighbors



3. Predict graph context and label using aggregated information

William L Hamilton et al., Inductive Representation Learning on Large Graphs, NIPS 2017

Graph Neural Networks

The GraphSAGE model introduces a spatial based filter, which is based on aggregating information from neighboring nodes. For a single node V_i, the process to generate its new features can be formulated as:

$$\begin{aligned} \mathcal{N}_S(v_i) &= SAMPLE(\mathcal{N}(v_i), S) \\ \mathbf{f}'_{\mathcal{N}_S(v_i)} &= AGGREGATE(\{\mathbf{F}_j, \forall v_j \in \mathcal{N}_S(v_i)\}) \\ \mathbf{F}'_i &= \sigma\left(\left[\mathbf{F}_i, \mathbf{f}'_{\mathcal{N}_S(v_i)}\right]\mathbf{\Theta}\right) \end{aligned} \tag{1}$$

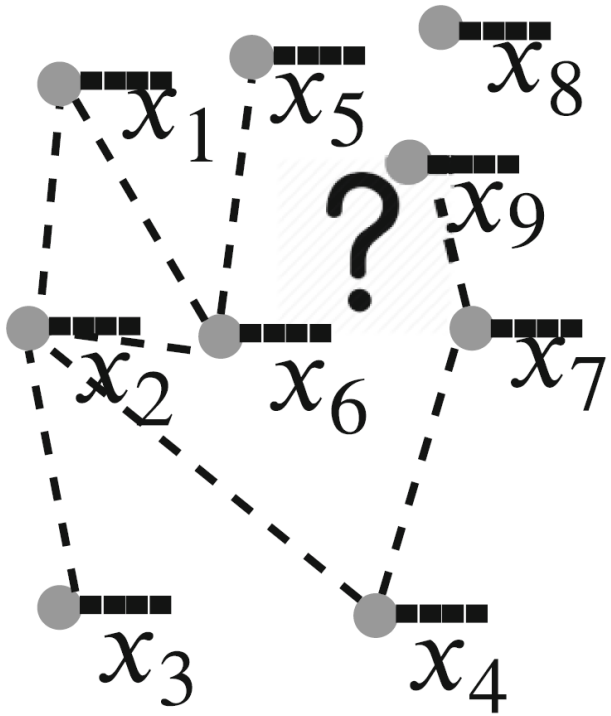
where $SAMPLE()$ is a function that takes a set as input and randomly samples S elements from the input as out, $AGGREGATE()$ is a function to combine the information from the neighboring nodes, and $[\cdot, \cdot]$ is the concatenation operation.

William L Hamilton et al., Inductive Representation Learning on Large Graphs, NIPS 2017

Outline

Latent Graph Inference

Given only the node features, how can we infer an underlying latent graph that optimally models the relationships between nodes?



Jianglin Lu et al. Latent Graph Inference with Limited Supervision. NeurIPS, 2023

Latent Graph Inference

Latent Graph Inference (LGI)

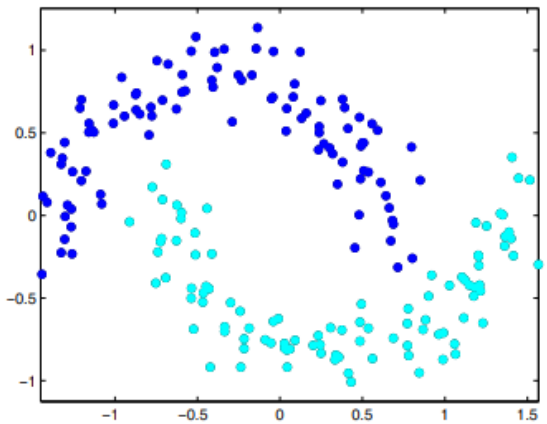
Given a graph $\mathcal{G}(\mathcal{V}, \mathbf{X})$ containing n nodes $\mathcal{V} = \{V_1, \ldots, V_n\}$ and a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ with each row $\mathbf{X}_{i:} \in \mathbb{R}^d$ representing the d-dimensional attributes of node V_i, latent graph inference aims to simultaneously learn the underlying graph topology encoded by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and the discriminative d'-dimensional node representations $\mathbf{Z} \in \mathbb{R}^{n \times d'}$ based on $\mathbf{X}$, where the learned $\mathbf{A}$ and $\mathbf{Z}$ are jointly optimal for certain downstream tasks $\mathcal{T}$ given a specific loss function $\mathcal{L}$.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision. NeurIPS, 2023

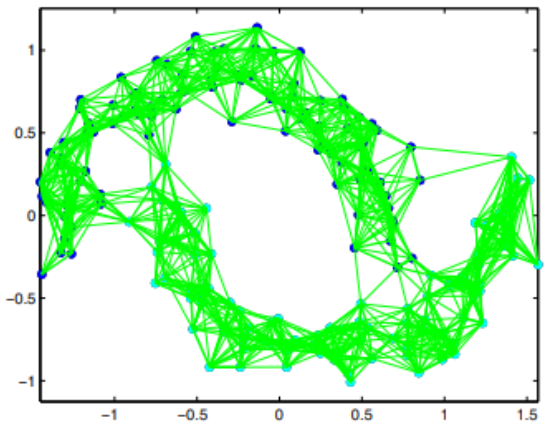
Outline

Clustering

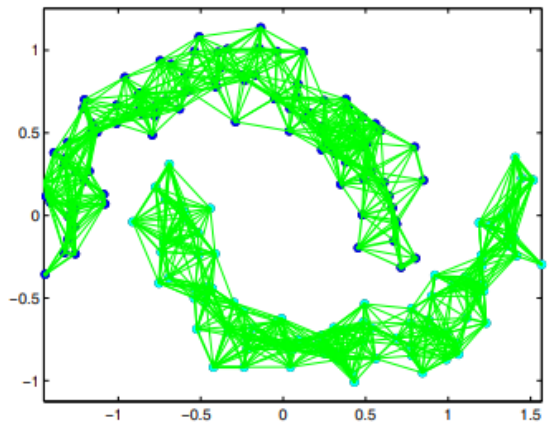
Clustering partitions the data points into different groups such that the objects in the same group have high similarity to each other.



(a) Original Data



(b) Learnt Graph by Eq.(5)

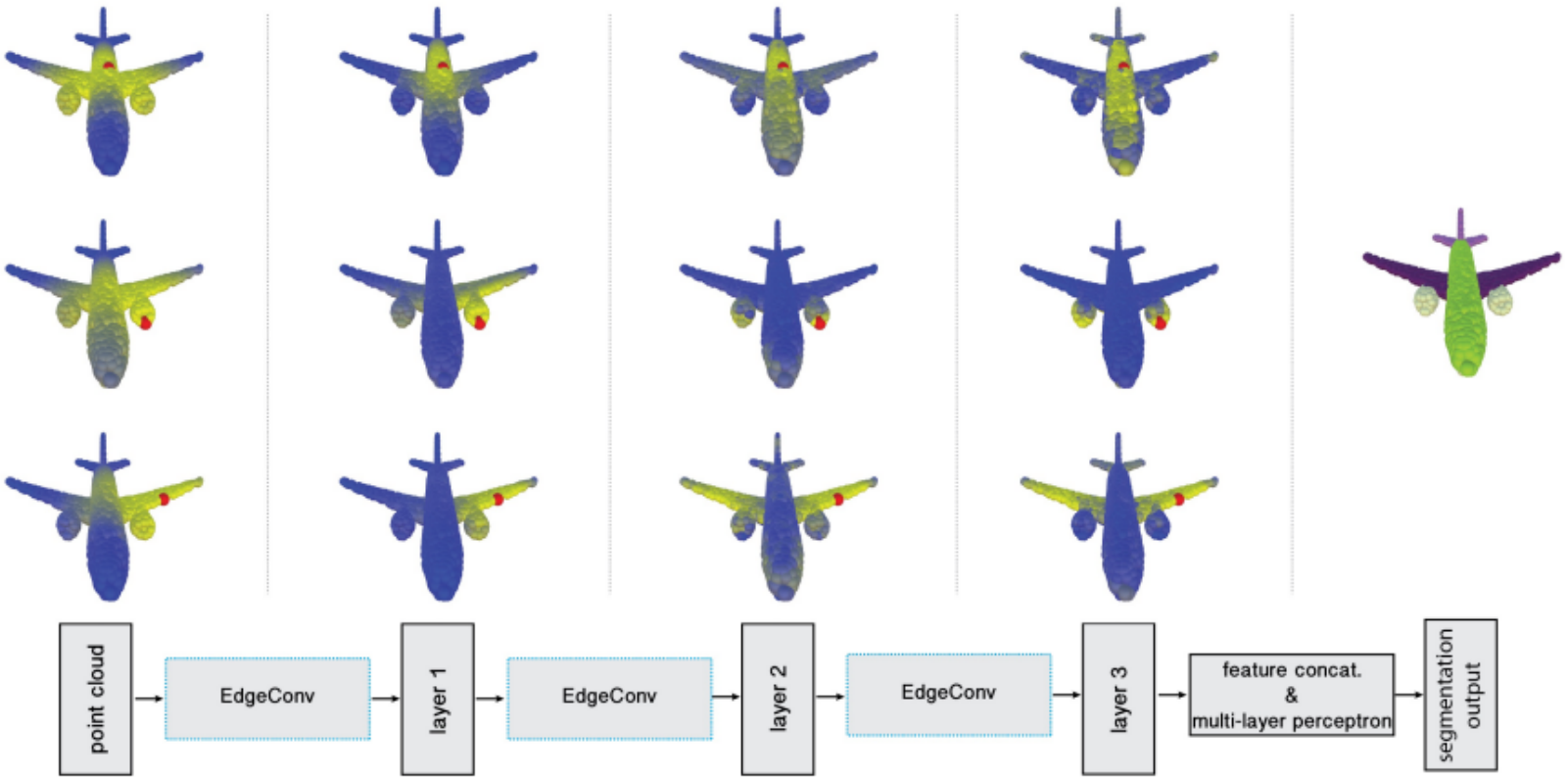


(c) Learnt Graph by Eq.(7) (CAN)

Feiping Nie et al. Clustering and Projected Clustering with Adaptive Neighbors, KDD 2014

Point Cloud Segmentation

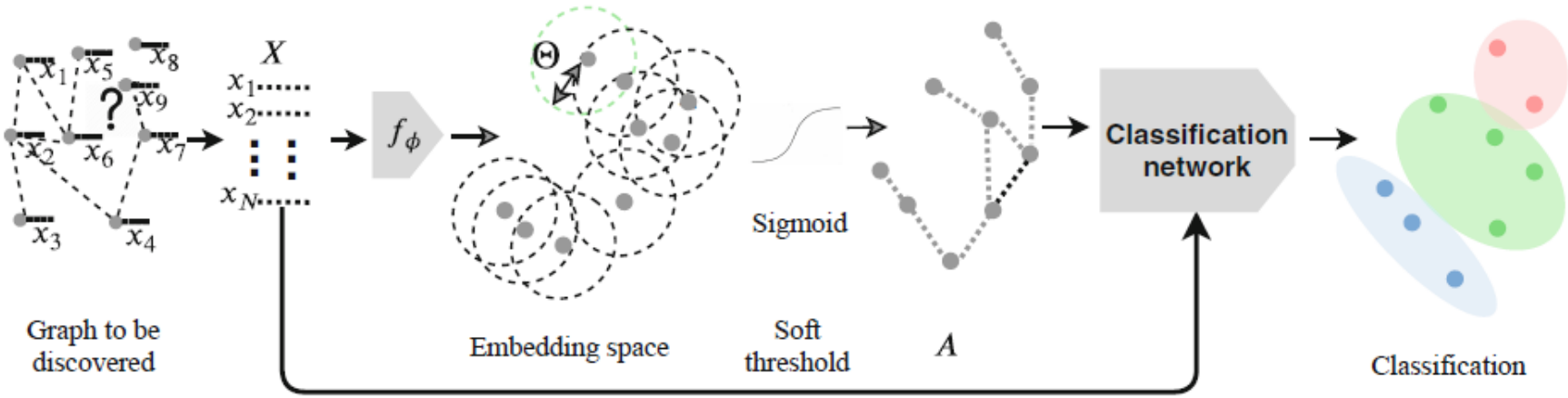
Point clouds, or scattered collections of points in 2D or 3D, are arguably the simplest shape representation.



Yue Wang et al. Dynamic Graph CNN for Learning on Point Clouds, ACM Trans. Graph., 2019

Disease Prediction

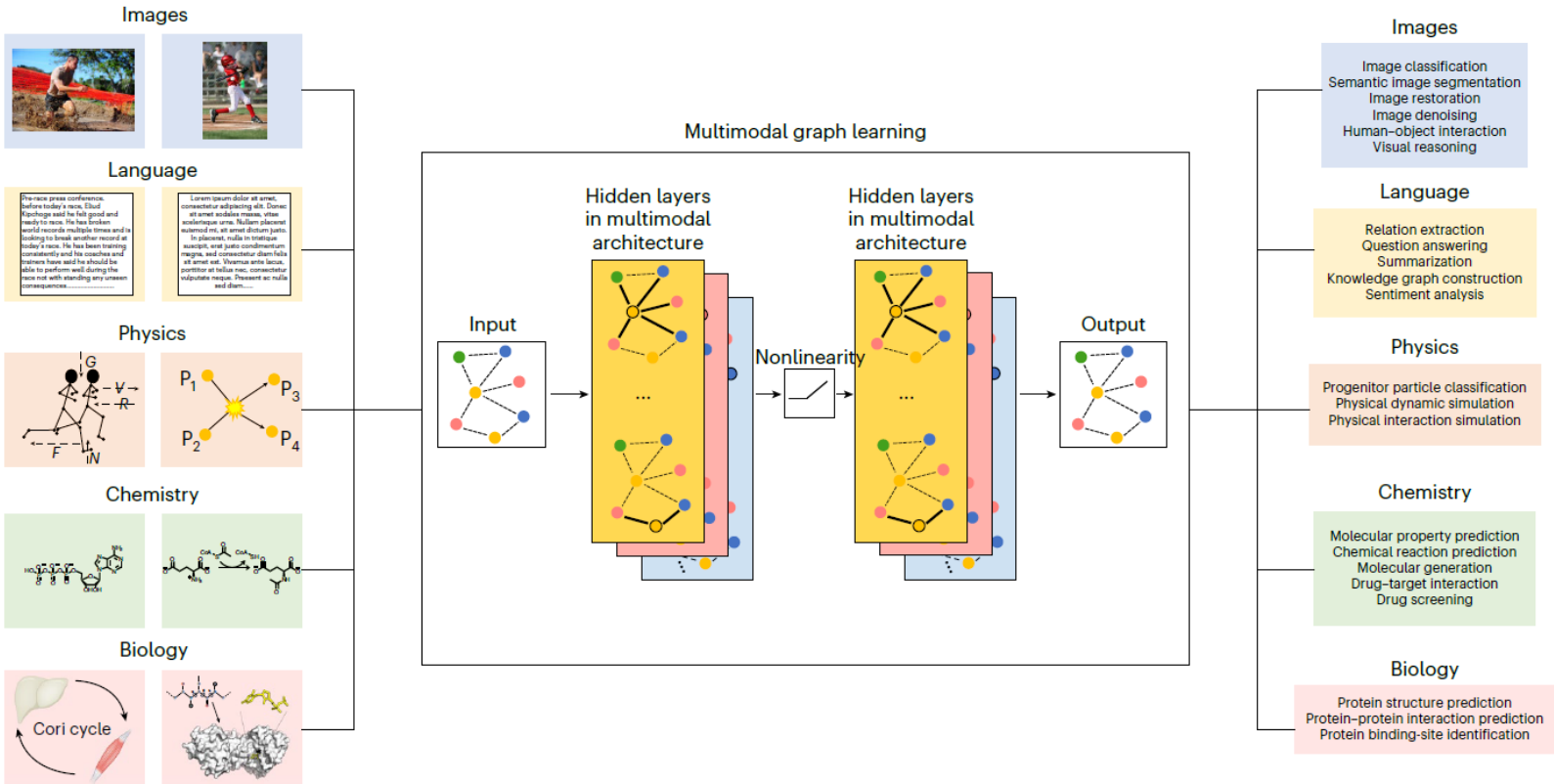
In the domain of computer aided diagnosis (CADx), it is possible to learn a single, optimal graph towards the downstream task of disease classification.



Luca Cosmo et al. Latent-Graph Learning for Disease Prediction, MICCAI., 2020

Multimodal Graph Learning

Artificial intelligence for graphs has achieved remarkable success in modelling complex systems, ranging from dynamic networks in biology to interacting particle systems in physics.



Yasha Ektefaie et al. Multimodal Learning with Graphs, Nat Mach Intell., 2023

Outline

Outline

Structured Optimal Graph

Basic Idea: In general, closer samples are likely to have larger probability to be connected. Thus, $\mathbf{A}_{ij}$ is inversely proportional to the distance between $\mathbf{X}_i$ and $\mathbf{X}_j$. Therefore, determining the value of the probability $\mathbf{A}_{ij}$ can be seen as solving:

$$\min_{\mathbf{A}} \sum_{i,j} \left(\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|_2^2 \mathbf{A}_{ij} + \alpha \mathbf{A}_{ij}^2 \right) \tag{2}$$
$$\text{s.t. } \forall i, \mathbf{A}_{i:}\mathbf{1} = 1, \quad 0 \leq \mathbf{A}_{ij} \leq 1$$

where the square of Euclidean distance $\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|_2^2$ is used for simplicity and the regularization term $\alpha \mathbf{A}_{ij}^2$ is used to avoid the trivial solution.

Feiping Nie et al. Unsupervised Feature Selection with Structured Graph Optimization. AAAI, 2016

Structured Optimal Graph

Structured Optimal Graph: The optimal similarity matrix should have exact c connected components, where c is the number of cluster:

$$\min_{\mathbf{A}} \sum_{i,j} \left(\|\mathbf{X}_{i:} - \mathbf{X}_{j:}\|_2^2 \mathbf{A}_{ij} + \alpha \mathbf{A}_{ij}^2 \right) \tag{3}$$

$$\text{s.t. } \forall i, \mathbf{A}_{i:}\mathbf{1} = 1, \quad 0 \leq \mathbf{A}_{ij} \leq 1, \ \text{rank}\left(\mathbf{L}_A\right) = n - c$$

where $\mathbf{L}_A$ is the Laplacian matrix of $\mathbf{A}$ and rank $(\mathbf{L}_A)$is the rank of $\mathbf{L}_A$.

It can be proved that if rank $(\mathbf{L}_A) = n - c$, the similarity matrix $\mathbf{A}$ will contain exact c connected components.

Feiping Nie et al. Unsupervised Feature Selection with Structured Graph Optimization. AAAI, 2016

Outline

Block Diagonal Representation

K-Block Diagonal Structure

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_k \end{bmatrix}, \mathbf{A}_i \in \mathbb{R}^{n \times n} \tag{4}$$

Canyi Lu et al. Subspace Clustering by Block Diagonal Representation. TPAMI, 2018

Block Diagonal Representation

Illustrations of three interesting structures of matrix: sparse, low-rank and block diagonal matrices.

(a) sparse (b) low-rank (c) block diagonal

Canyi Lu et al. Subspace Clustering by Block Diagonal Representation. TPAMI, 2018

Block Diagonal Representation

Key Idea: If the affinity matrix is block diagonal, i.e., the between-cluster affinities are all zeros, one may achieve perfect data clustering by using spectral clustering:

$$\min_{\mathbf{A}} \frac{1}{2}\left|\left|\mathbf{X}^T - \mathbf{X}^T\mathbf{A}\right|\right|^2 + \gamma\|\mathbf{A}\|_{[k]}, \quad (5)$$
$$\text{s.t. } \operatorname{diag}(\mathbf{A}) = 0, \mathbf{A} \geq 0, \mathbf{A} = \mathbf{A}^\top$$

where

$$\|\mathbf{A}\|_{[k]} = \sum_{i=n-k+1}^{n} \lambda_i\left(\mathbf{L}_\mathbf{A}\right) \quad (6)$$

and $\lambda_i\left(\mathbf{L}_\mathbf{A}\right)$ are the eigenvalues of $\mathbf{L}_\mathbf{A}$ in the decreasing order.

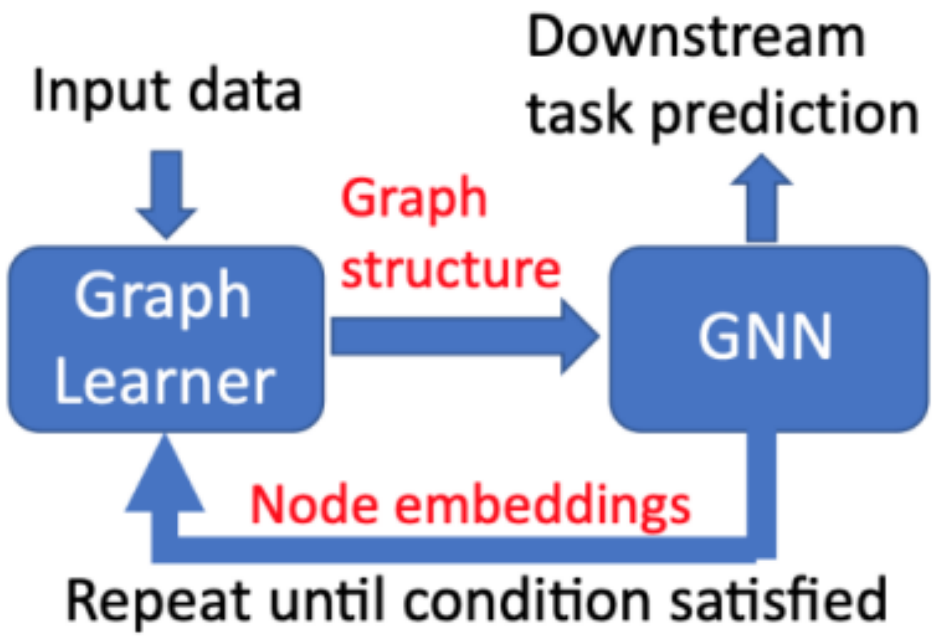
Canyi Lu et al. Subspace Clustering by Block Diagonal Representation. TPAMI, 2018

Outline

Outline

Iterative Deep Graph Learning

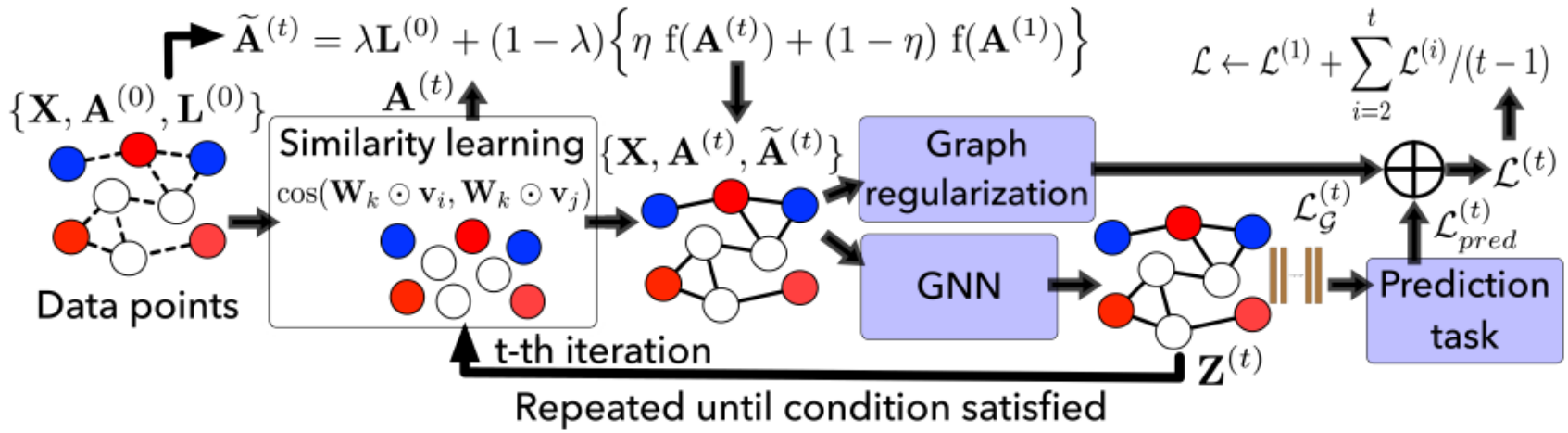
Key Idea: IDGL learns a better graph structure based on better node embeddings, and vice versa (i.e., better node embeddings based on a better graph structure).



Yu Chen et al. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings, 2020

Iterative Deep Graph Learning

IDGL: uses multi-head self-attention with epsilon-neighborhood sparsification for constructing a graph, and optimizes a joint loss combining both task-specific prediction loss and graph regularization loss.

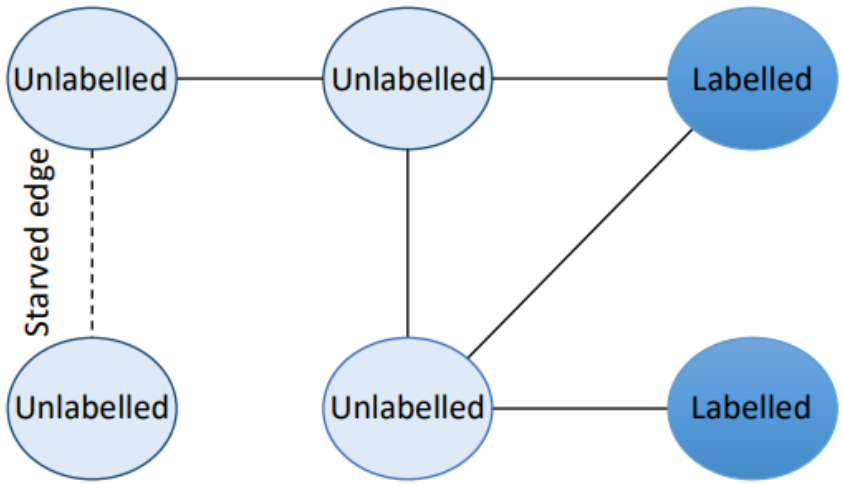


Yu Chen et al. Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings, 2020

Outline

SLAPS

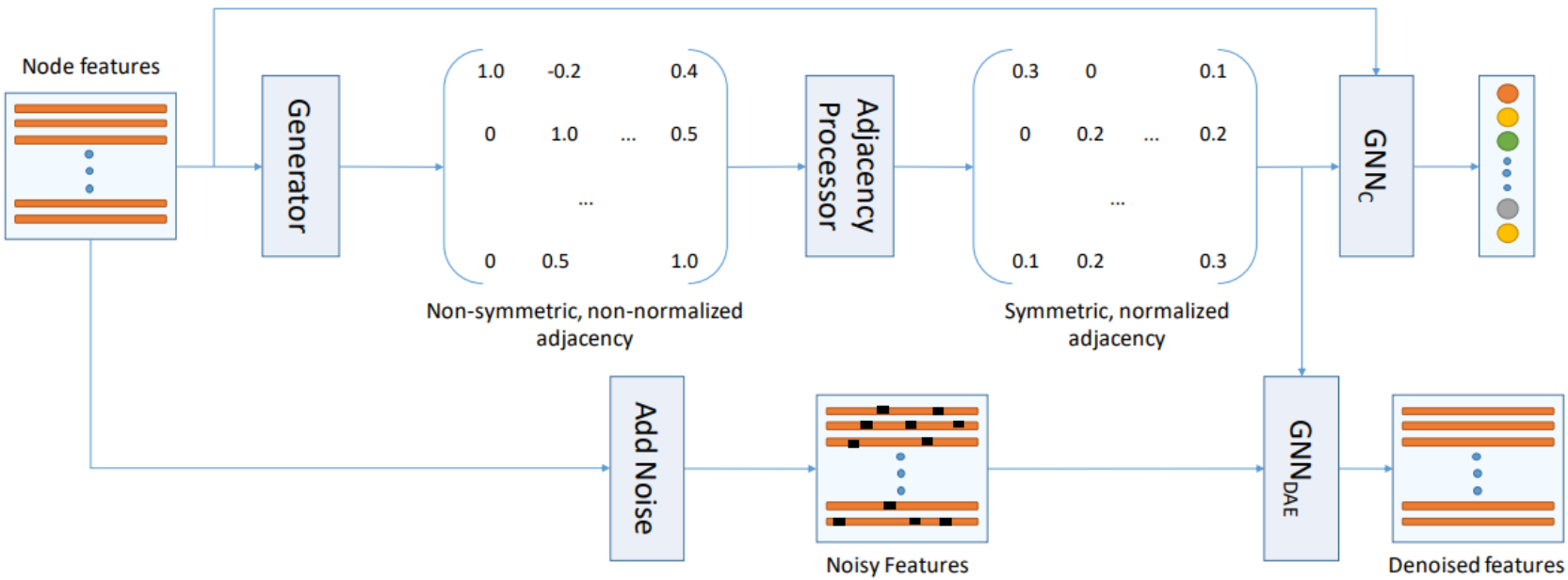
Supervision Starvation: Starved edges exist in existing LGI methods. These edges are problematic as the predictions at the test time depend on these edges. If their values are learned without enough supervision, the model may make poor predictions at the test time.



Bahare Fatemi et al. SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks, NeurIPS, 2021

SLAPS

Key Idea: take a learning-based approach based on self-supervision. The learned graph structure is used for both the classification task and a denoising task on the node features. The self-supervised task encourages the model to learn a structure that is suitable for predicting the node features.



Bahare Fatemi et al. SLAPS: Self-Supervision Improves Structure Learning for Graph Neural Networks, NeurIPS, 2021

Outline

Outline

Latent Graph Inference with Limited Supervision

Why Supervision Starvation Happens? In fact, the SS problem is caused by a common and necessary post-processing operation known as graph sparsification, which adjusts the initial dense graph to a sparse one:

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{A}_{ij}, & \text{if } \mathbf{A}_{ij} \in \text{top-}\kappa(\mathbf{A}_{i:}) \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where top-$\kappa(\mathbf{A}_{i:})$ denotes the set of the top κ values in $\mathbf{A}_{i:}$. After this sparsification operation, a significant number of edge weights are directly erased.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

How Many Nodes Suffer from This Problem?

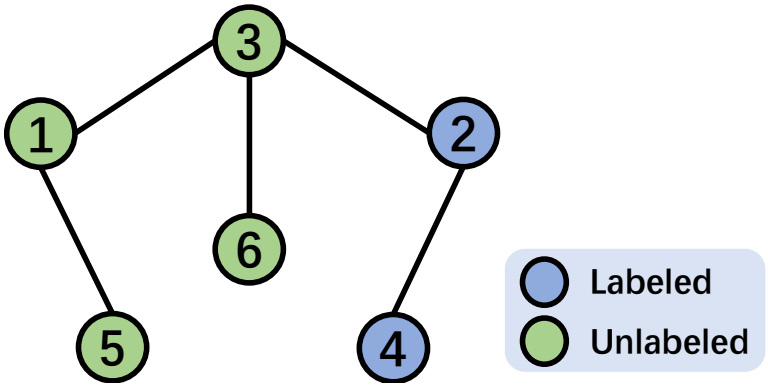
k-hop Starved Node

Given a graph $\mathcal{G}(\mathcal{V}, \mathbf{X})$ consisting of n nodes $\mathcal{V} = \{V_1, \ldots, V_n\}$ and the corresponding node features $\mathbf{X}$, for a k-layer graph neural network $\texttt{GNN}_k(\mathbf{X}; \mathbf{\Theta})$ with network parameters $\mathbf{\Theta}$, the unlabeled node V_i is a k-hop starved node if, for $\forall \kappa \in \{1, \ldots, k\}$, $\forall V_j \in \mathcal{N}_\kappa(i)$, where $\mathcal{N}_\kappa(i)$ is the set of κ-hop neighbors of V_i, V_j is unlabeled. Specifically, 0-hop starved nodes are defined as the unlabeled nodes.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

A Toy Example: Given a 2-layer GNN, node 5 is a 2-hop starved node.

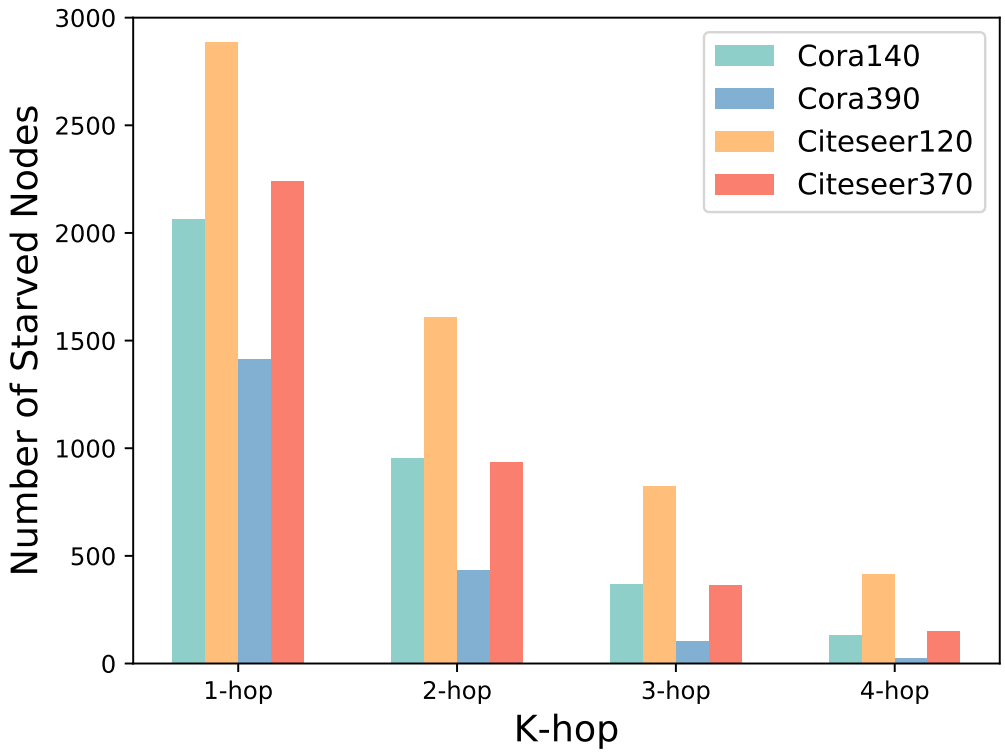


$$\mathbf{A}: \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

Real-World Data: The more labeled nodes, the smaller the number of starved nodes. This is natural because the more labeled nodes, the greater the probability that a node will connect to a labeled node. Moreover, the number of k-hop starved nodes decreases as k increases.



How to Identify Starved Nodes?

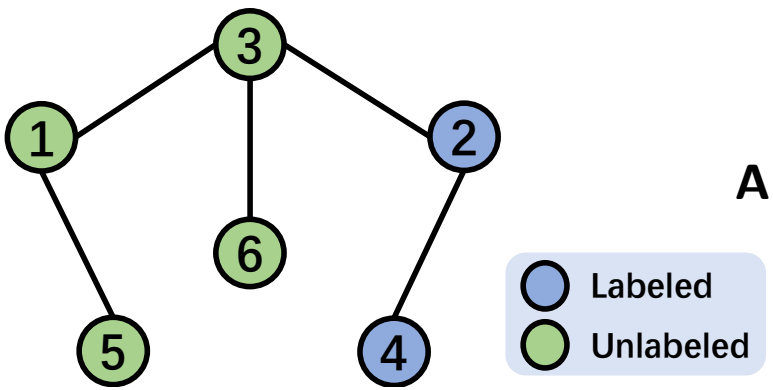
Identification of Starved Nodes

Given a sparse adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with self-connections generated on graph $\mathcal{G}(\mathcal{V}, \mathbf{X})$ by a latent graph inference model with a k-layer graph neural network $\mathtt{GNN}_k(\mathbf{X}; \mathbf{\Theta})$, the node V_i is a k-hop starved node, if $\exists j \in \{1, \ldots, n\}$, such that $[\mathbb{1}_{\mathbb{R}^+}(\mathbf{A})]_{ij}^k = 1$, and for $\forall j \in \{j \mid [\mathbb{1}_{\mathbb{R}^+}(\mathbf{A})]_{ij} = 1 \cup [\mathbb{1}_{\mathbb{R}^+}(\mathbf{A})]_{ij}^2 = 1 \cup \ldots \cup [\mathbb{1}_{\mathbb{R}^+}(\mathbf{A})]_{ij}^k = 1\}$, V_j is unlabeled.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

Illustration: Since nodes V_2 and V_4 are labeled, we identify the 1-hop starved nodes as $\{V_1, V_5, V_6\}$ (self-connections are not considered when defining k-hop neighbors.).



$$\mathbf{A}: \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{l} (V_3, V_5) \\ (V_3, V_4) \\ (V_1, V_2, V_6) \\ (V_2) \\ (V_1) \\ (V_3) \end{array}$$

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

Illustration: We can identify 2-hop starved nodes from the set $\{V_1, V_5, V_6\}$ as $\{V_5\}$.

$$\mathbf{A}^2: \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 3 & 1 & 2 & 0 & 2 & 1 \\ 1 & 3 & 2 & 2 & 0 & 1 \\ 2 & 2 & 4 & 1 & 1 & 2 \\ 0 & 2 & 1 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 & 2 & 0 \\ 1 & 1 & 2 & 0 & 0 & 2 \end{bmatrix} \end{array} \begin{array}{l} \\ (V_2, V_6) \\ (V_1, V_6) \\ (V_4, V_5) \\ (V_3) \\ (V_3) \\ (V_1, V_2) \end{array}$$

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

CUR Decomposition Makes Better Solution:

CUR Decomposition

Given $\mathbf{Q} \in \mathbb{R}^{n \times m}$ of rank $\rho = \texttt{rank}(\mathbf{Q})$, rank parameter $k < \rho$, and accuracy parameter $0 < \varepsilon < 1$, construct column matrix $\mathbf{C} \in \mathbb{R}^{n \times c}$ with c columns from $\mathbf{Q}$, row matrix $\mathbf{R} \in \mathbb{R}^{r \times m}$ with r rows from $\mathbf{Q}$, and intersection matrix $\mathbf{U} \in \mathbb{R}^{c \times r}$ with c, r, and $\texttt{rank}(\mathbf{U})$ being as small as possible, in oder to reconstruct $\mathbf{Q}$ within relative-error:

$$||\mathbf{Q} - \mathbf{CUR}||_F^2 \leq (1 + \varepsilon)||\mathbf{Q} - \mathbf{Q}_k||_F^2. \tag{8}$$

Here, $\mathbf{Q}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \in \mathbb{R}^{n \times m}$ is the best rank k matrix obtained via the singular value decomposition (SVD) of $\mathbf{Q}$.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

CUR Decomposition Makes Better Solution:

Identification of Starved Nodes

Given a sparse adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with self-connections generated on graph $\mathcal{G}(\mathcal{V}, \mathbf{X})$, construct $\mathbf{C} = \mathbf{A}[:, col\_mask] \in \mathbb{R}^{n \times c}$, where $col\_mask \in \{0,1\}^n$ contains only c positive values corresponding to c labeled nodes, and $\mathbf{R} = \mathbf{A}[row\_mask, :] \in \mathbb{R}^{r \times n}$ with $row\_mask = \mathbb{1}_{\mathbb{R}^-}(\mathbf{C}\mathbb{1}_c) \in \{0,1\}^n$. Then, (a) $\mathbf{U} = \mathbf{A}[row\_mask, col\_mask] = \mathbf{0} \in \mathbb{R}^{r \times c}$, where $\mathbf{0}$ is a zero matrix, (b) the set of 1-hop starved nodes $\mathtt{Set}_1(r) = \{V_i | i \in \mathtt{RM}_+\}$, where $\mathtt{RM}_+ \in \mathbb{N}^r$ indicates the set of indexes of positive elements from row_mask, and (c) for each $i \in \mathtt{RM}_+$, V_i is a 2-hop starved node if, for $\forall j$ satisfying $[\mathbb{1}_{\mathbb{R}^+}(\mathbf{R})]_{ij} = 1$, $j \in \mathtt{RM}_+$.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

Latent Graph Inference with Limited Supervision

Illustration: Based on the $\mathbf{C}, \mathbf{U}, \mathbf{R}$ matrices, we can determine that $row\_mask = [1,0,0,0,1,1]^\top$, $\mathtt{RM}_+ = \{1,5,6\}$, the 1-hop starved nodes are $\{V_1, V_5, V_6\}$, and the 2-hop starved node is V_5.

$$\mathbf{C}: \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array} \quad \mathbf{R}: \begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \quad \mathbf{U}: \begin{array}{c} \\ 1 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array}$$

Latent Graph Inference with Limited Supervision

How to Eliminate Starved Nodes? After identification, we can reduce the starved nodes by rebuilding the corrupted affinities. Specifically, we rebuild the intersection matrix $\mathbf{U}$ to ensure that the reconstructed $\widetilde{\mathbf{U}} \neq \mathbf{0}$:

$$\widetilde{\mathbf{A}} = \mathbf{A} + \alpha \mathbf{B} = \mathbf{A} + \alpha \Gamma \left(\widetilde{\mathbf{U}}, n \right), \tag{9}$$

where function $\Gamma(\widetilde{\mathbf{U}}, n)$ extends the matrix $\widetilde{\mathbf{U}} \in \mathbb{R}^{r \times c}$ to an $n \times n$ matrix by padding $n - r$ rows of zeros and $n - c$ columns of zeros in the corresponding positions.

Jianglin Lu et al. Latent Graph Inference with Limited Supervision, NeurIPS, 2023

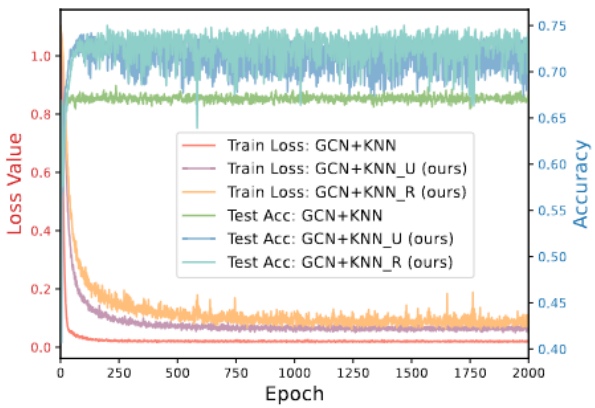
Experiments

Table 1: Test accuracy (%) of the baselines (M) and our CUR extension versions (M_U and M_R) on various datasets with different labeling rates (marked in **bold**), where "OOM" indicates out of memory. The highest and second highest results are marked in red and blue, respectively.

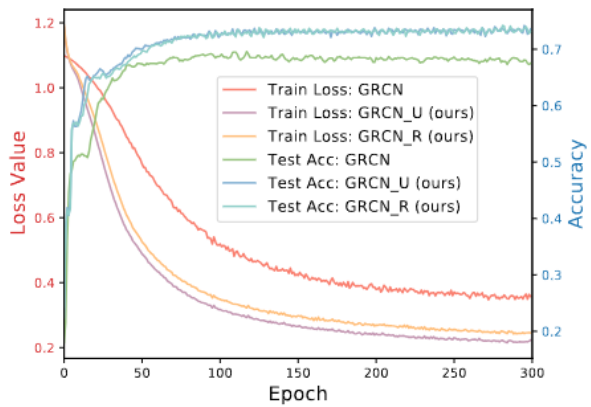
Models / datasets	ogbn-arxiv	Cora390	Cora140	Citeseer370	Citeseer120	Pubmed
# of labeled / all nodes	90941/169343	390/2708	140/2708	370/3327	120/3327	60/19717
Labeling rate	**53.70%**	**14.40%**	**5.17%**	**11.12%**	**3.61%**	**0.30%**
GCN+KNN	55.15 ± 0.11	72.82 ± 0.39	67.94 ± 0.29	73.28 ± 0.23	69.68 ± 0.53	68.66 ± 0.05
GCN+KNN_U (ours)	55.82 ± 0.11	72.82 ± 0.21	68.18 ± 0.44	73.68 ± 0.10	69.74 ± 0.54	74.12 ± 0.32
GCN+KNN_R (ours)	55.86 ± 0.10	72.92 ± 0.28	68.12 ± 0.48	73.66 ± 0.14	69.90 ± 0.68	74.78 ± 0.17
GCN&KNN	OOM	72.16 ± 0.54	68.76 ± 1.20	77.28 ± 0.64	68.64 ± 1.14	OOM
GCN&KNN_U (ours)	OOM	73.04 ± 0.20	70.16 ± 0.91	78.40 ± 0.44	70.52 ± 1.04	OOM
GCN&KNN_R (ours)	OOM	73.20 ± 0.25	70.24 ± 0.97	78.48 ± 0.30	69.48 ± 0.77	OOM
IDGL [4]	OOM	74.00 ± 0.38	70.74 ± 0.50	71.30 ± 0.17	69.24 ± 0.19	OOM
IDGL_U (ours)	OOM	74.54 ± 0.52	70.82 ± 0.49	72.46 ± 0.14	69.32 ± 0.39	OOM
IDGL_R (ours)	OOM	74.48 ± 0.47	71.14 ± 0.22	72.56 ± 0.12	69.86 ± 0.50	OOM
LCGS [13]	OOM	72.02 ± 0.37	69.88 ± 0.66	73.84 ± 0.83	72.30 ± 0.33	OOM
LCGS_U (ours)	OOM	72.18 ± 0.31	70.04 ± 0.80	74.18 ± 0.43	72.38 ± 0.43	OOM
LCGS_R (ours)	OOM	72.22 ± 0.45	70.14 ± 0.64	74.20 ± 0.36	72.40 ± 0.42	OOM
GRCN [46]	OOM	73.34 ± 0.27	68.86 ± 0.25	73.62 ± 0.23	71.24 ± 0.19	69.24 ± 0.20
GRCN_U (ours)	OOM	74.10 ± 0.25	69.44 ± 0.34	73.88 ± 0.34	71.54 ± 0.31	72.80 ± 0.99
GRCN_R (ours)	OOM	74.14 ± 0.22	69.56 ± 0.22	74.22 ± 0.13	71.64 ± 0.41	72.82 ± 1.03
SLAPS [10]	55.46 ± 0.12	76.62 ± 0.83	74.26 ± 0.53	74.32 ± 0.56	70.66 ± 0.97	74.86 ± 0.79
SLAPS_U (ours)	55.68 ± 0.09	76.94 ± 0.42	74.56 ± 0.21	74.82 ± 0.27	71.68 ± 0.47	76.74 ± 0.59
SLAPS_R (ours)	56.11 ± 0.15	76.82 ± 0.19	75.00 ± 0.49	74.90 ± 0.42	72.36 ± 0.49	77.12 ± 0.77

Latent Graph Inference with Limited Supervision

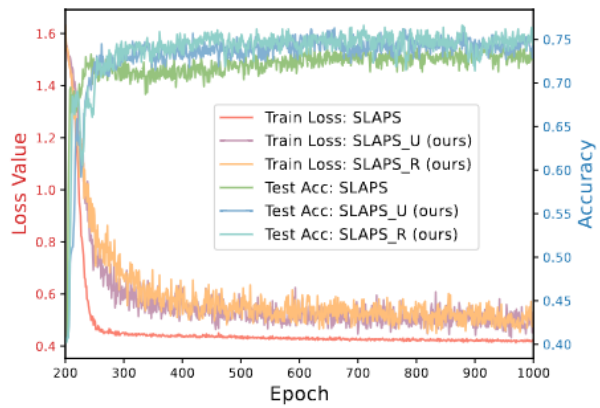
Experiments



(a) GCN+KNN



(b) GRCN



(c) SLAPS

The End!