ICML International Conference On Machine Learning

Commute Graph Neural Networks

Wei Zhuo[1,2], Han Yu[2], Guang Tan[1], Xiaoxiao Li[3,4]

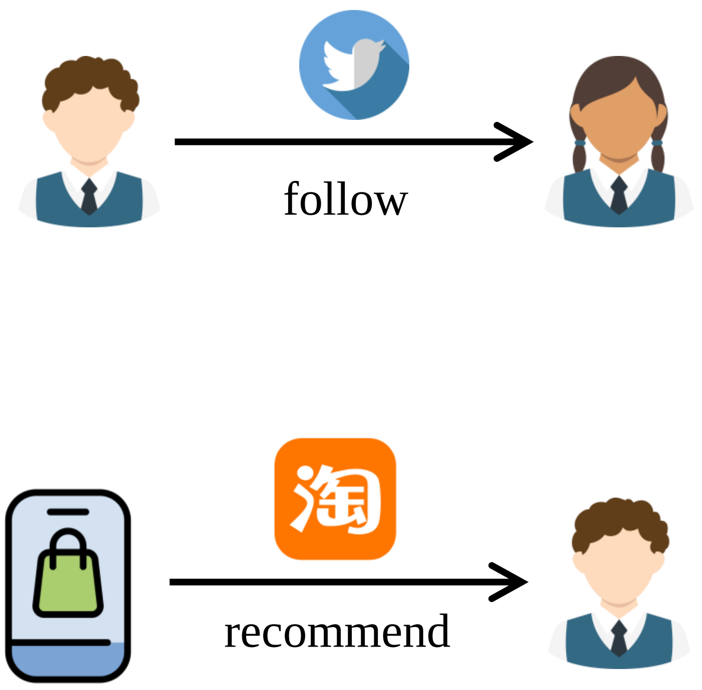
[1]Shenzhen Campus of Sun Yat-sen University

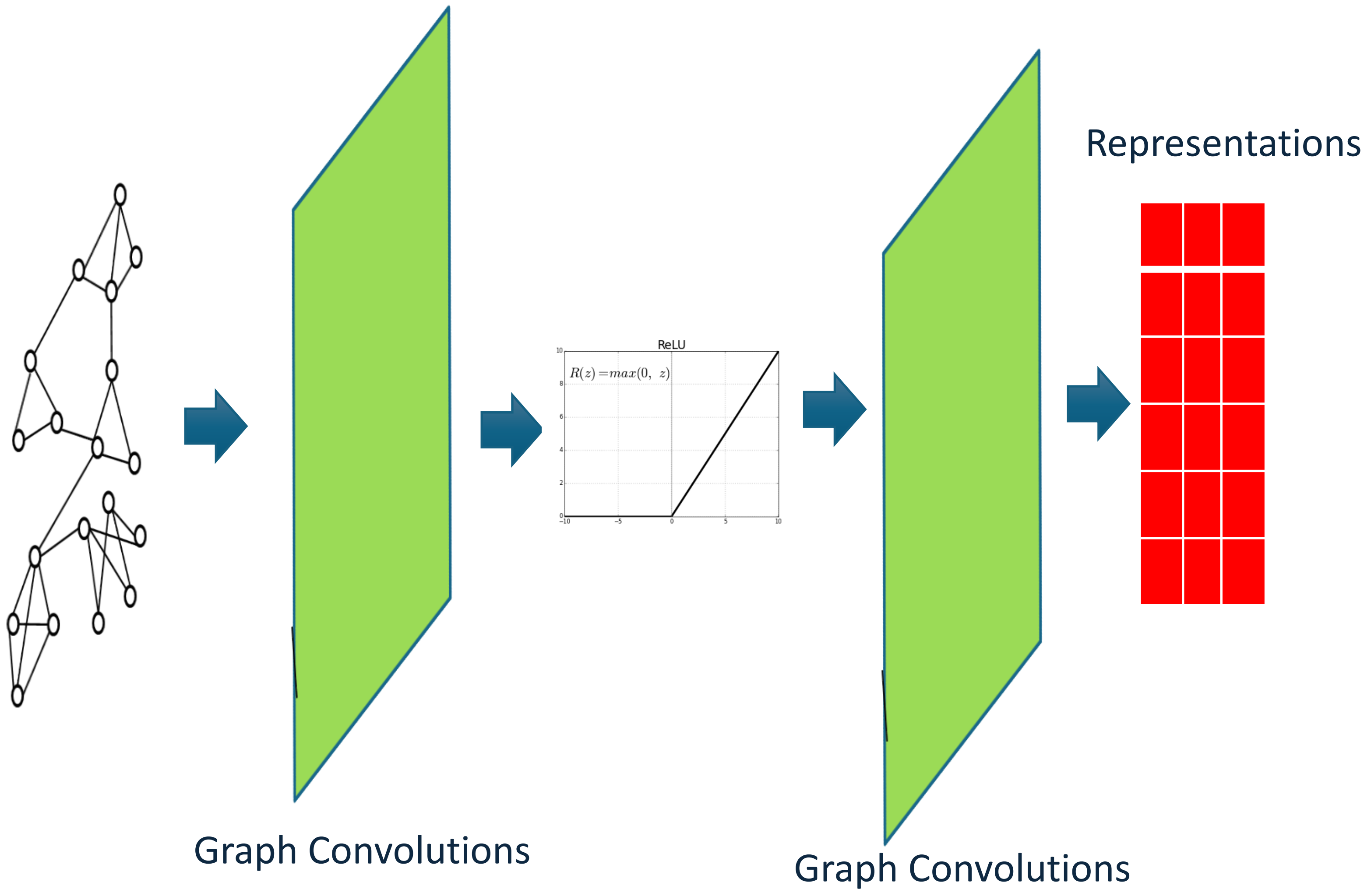
[2]Nanyang Technological University

[3]University of British Columbia, [4]Vector Institute

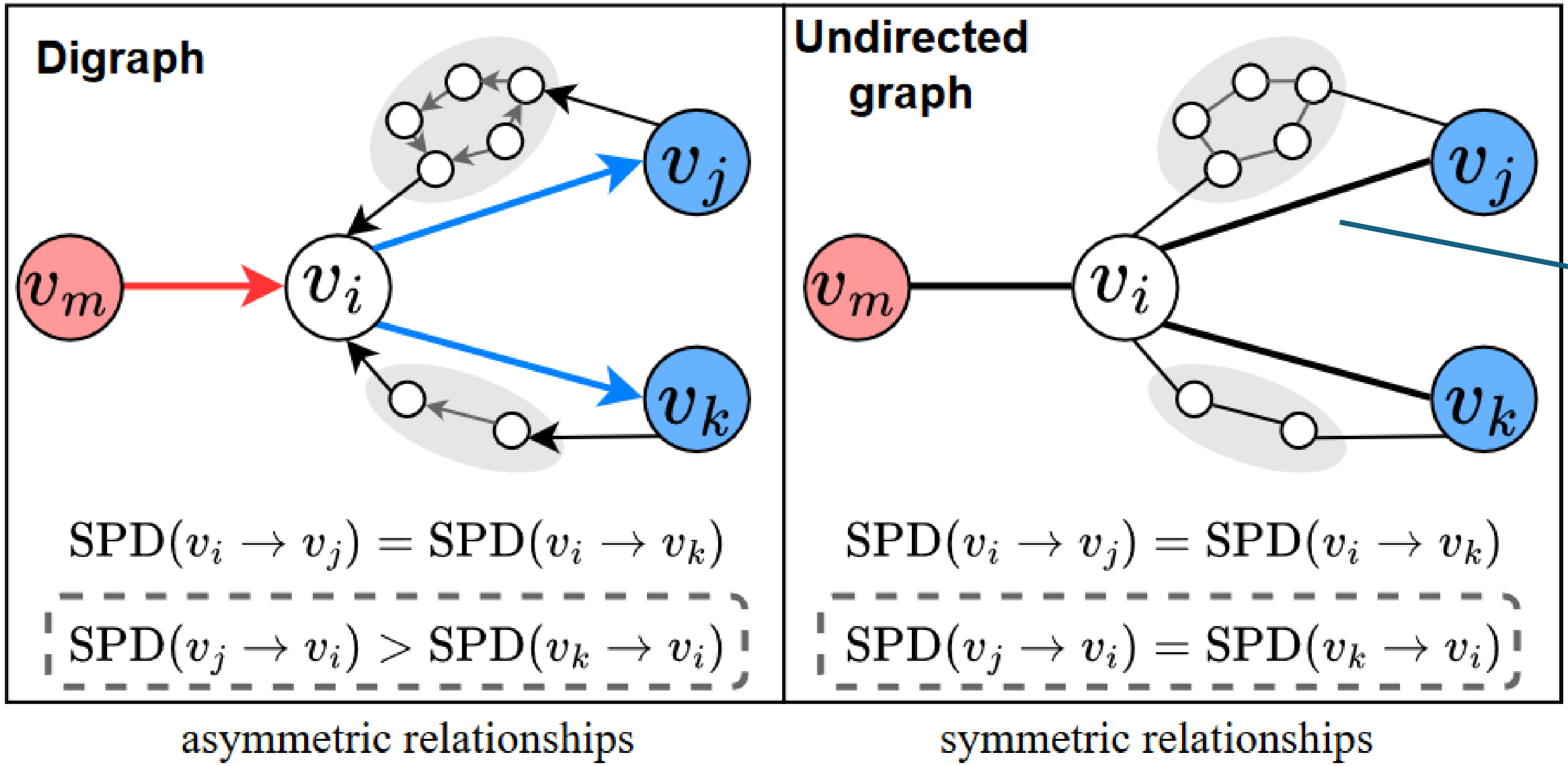
Problem formulation

Directed Graph (Digraph):





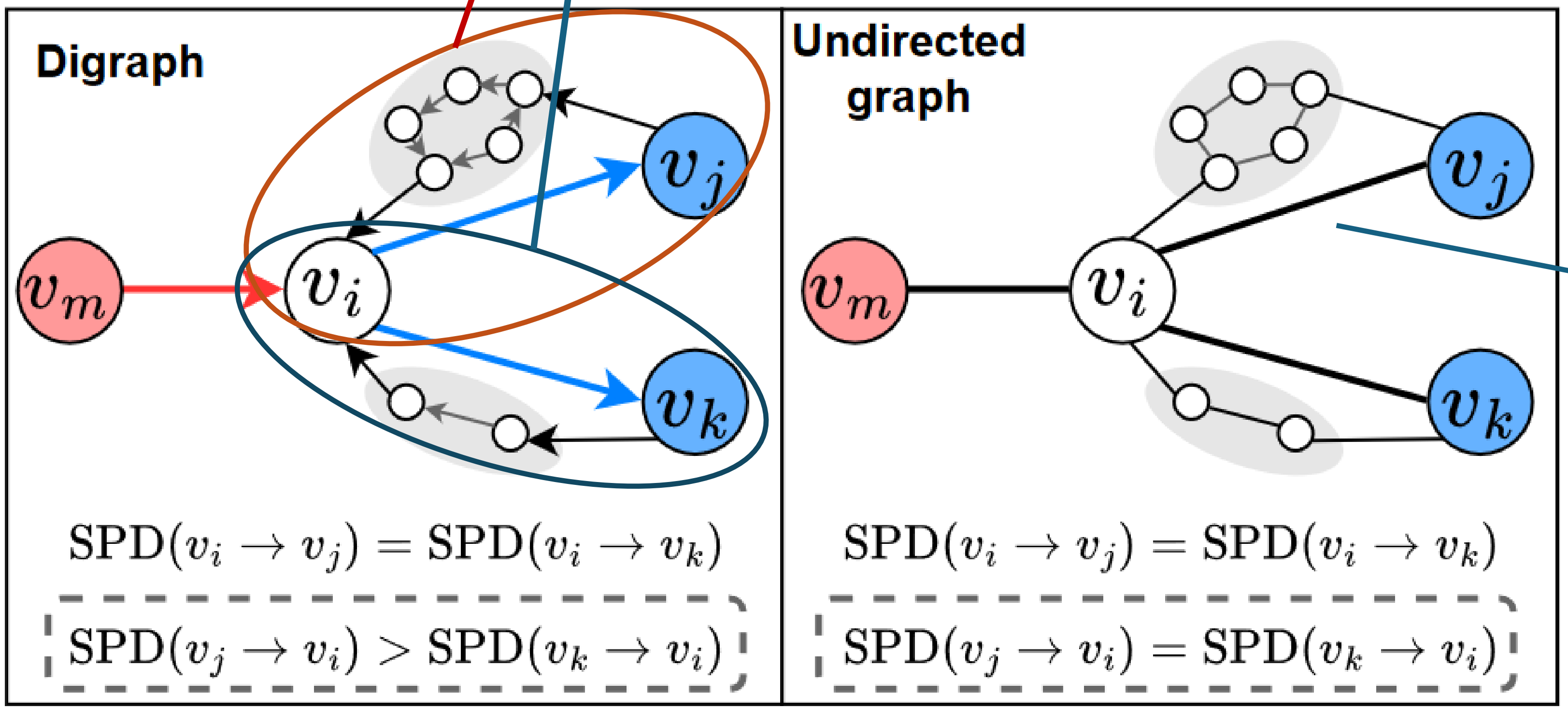
Relationships between nodes in a directed graph is more complex than that in undirected graphs.



Commute closed-loop between loop between v_i and v_j

$$\mathrm{Commute}(v_i, v_j) \neq \mathrm{Commute}(v_i, v_k)$$

Relationships between nodes in a directed graph is more complex than that in undirected graphs.

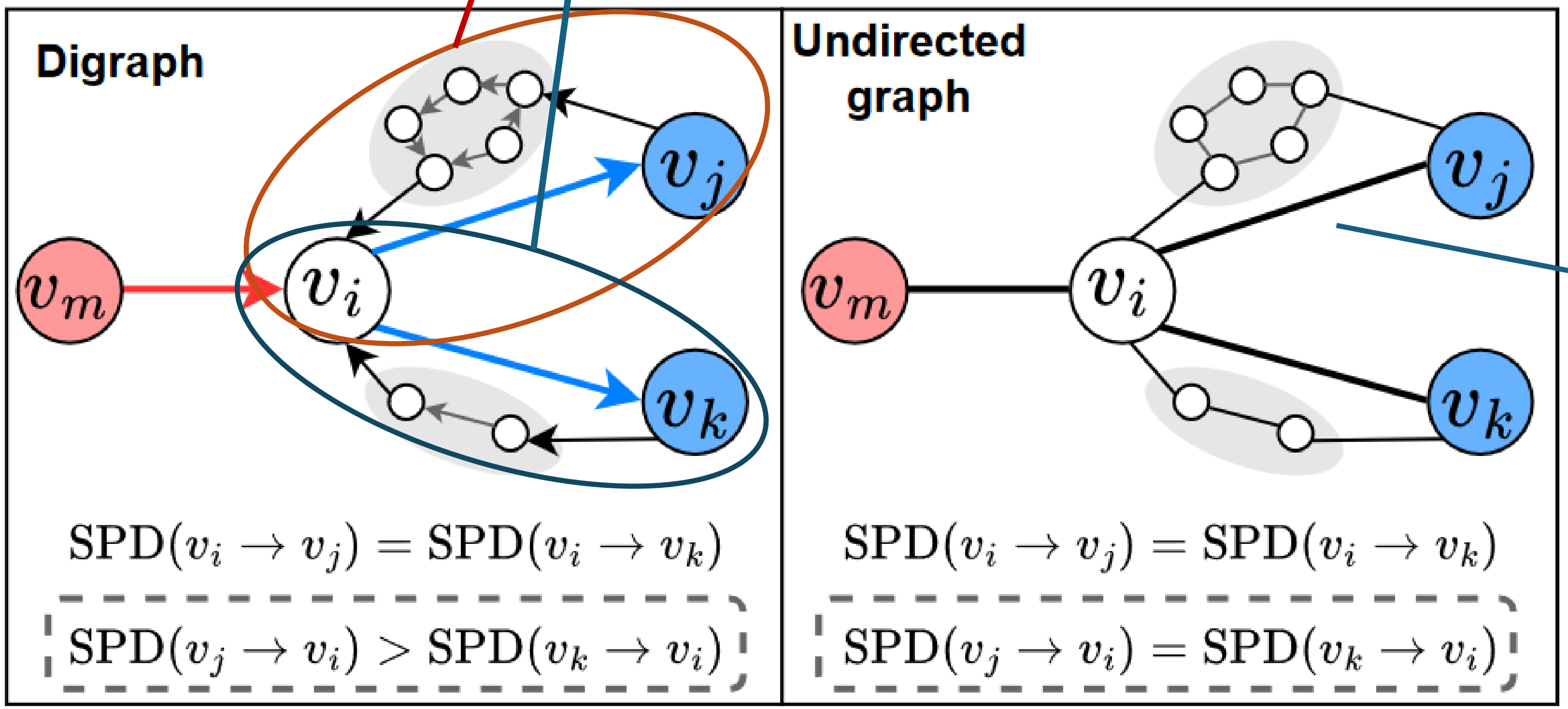


asymmetric relationships　　symmetric relationships

Commute closed-loop between loop between v_i and v_j

$$\text{Commute}(v_i, v_j) > \text{Commute}(v_i, v_k)$$

Relationships between nodes in a directed graph is more complex than that in undirected graphs.



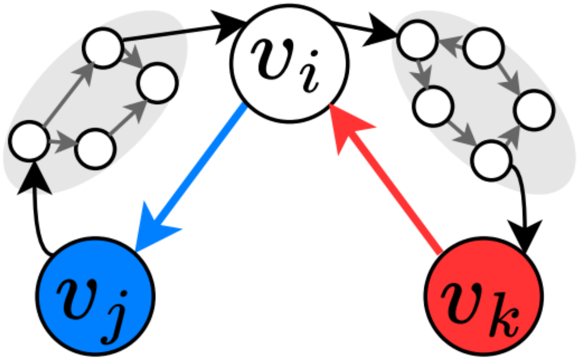
asymmetric relationships

symmetric relationships

Commute closed-loop between loop between v_i and v_j

The strength of the relationship between v_i and v_k is stronger than that between v_i and v_j.

Markov chain theory defines the expected length of the round-trip path between nodes using **commute time**, therefore, the goal of this study is to ensure that the relationships between node vector representations of GNNs can reflect the commute times between nodes.



Theorem 3.2. (Li & Zhang, 2012) *Given Assumption 3.1, the fundamental matrix* $\mathbf{Z}$ *defined in Eq. (2) converges to:*

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{J}\mathbf{\Pi})^{-1} - \mathbf{J}\mathbf{\Pi}, \tag{3}$$

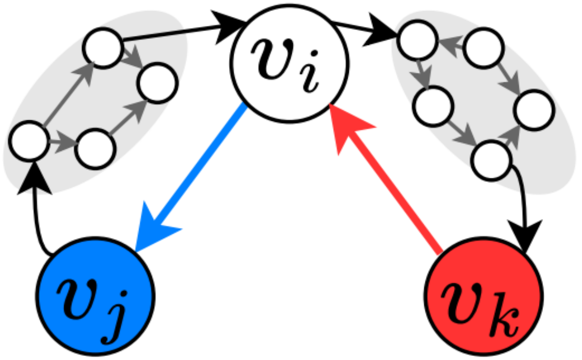
where $\mathbf{I}$ *is an identity matrix.*

The hitting time and commute time on G can then be expressed as $\mathbf{Z}$ (Aldous & Fill, 2002) as follows:

$$h(v_i, v_j) = \frac{\mathbf{Z}_{jj} - \mathbf{Z}_{ij}}{\pi_j}, \quad c(v_i, v_j) = h(v_i, v_j) + h(v_j, v_i). \tag{4}$$

[1] Li Y, Zhang Z-L. Digraph laplacian and the degree of asymmetry[J]. Internet Mathematics, 2012.

Markov chain theory defines the expected length of the round-trip path between nodes using **commute time**, therefore, the goal of this study is to ensure that the relationships between node vector representations of GNNs can reflect the commute times between nodes.



Theorem 3.2. *(Li & Zhang, 2012) Given Assumption 3.1, the fundamental matrix* $\mathbf{Z}$ *defined in Eq. (2) converges to:*

$$\mathbf{Z} = (\mathbf{I} - \mathbf{P} + \mathbf{J}\mathbf{\Pi})^{-1} - \mathbf{J}\mathbf{\Pi}. \tag{3}$$

Perron vector may not unique and meaningful

Inversion operation of a dense matrix

where $\mathbf{I}$ *is an identity matrix.*

The hitting time and commute time on G can then be expressed as $\mathbf{Z}$ (Aldous & Fill, 2002) as follows:

$$h(v_i, v_j) = \frac{\mathbf{Z}_{jj} - \mathbf{Z}_{ij}}{\pi_j}, \quad c(v_i, v_j) = h(v_i, v_j) + h(v_j, v_i). \tag{4}$$

[1] Li Y, Zhang Z-L. Digraph laplacian and the degree of asymmetry[J]. Internet Mathematics, 2012.

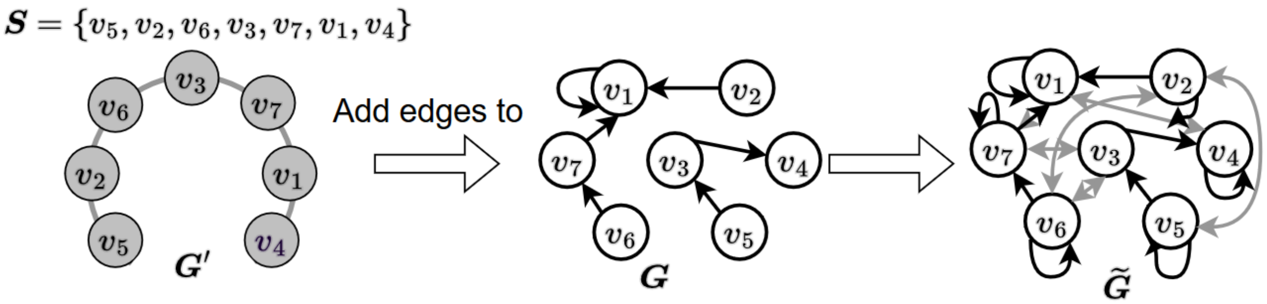
Problem 1： Perron vector may not unique and non-zero.

Solution： Similarity-based graph rewiring.

Theorem 1: If a graph is irreducible, then its Perron vector is non-zero. If a graph is aperiodic, then its Perron vector is unique.

Sufficient Conditions

Proposition 1: A strongly connected digraph, in which a directed path exists between every pair of vertices, is irreducible. A digraph with self-loops in each node is aperiodic.



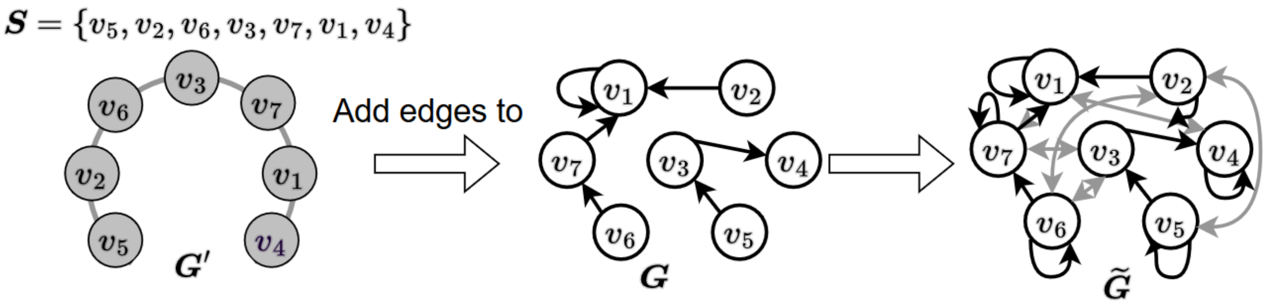
Problem 1： Perron vector may not unique and non-zero.

Solution： Similarity-based graph rewiring.

Theorem 1: If a graph is irreducible, then its Perron vector is non-zero. If a graph is aperiodic, then its Perron vector is unique.

Sufficient Conditions

Proposition 1: A strongly connected digraph, in which a directed path exists between every pair of vertices, is irreducible. A digraph with self-loops in each node is aperiodic.



Problem 2： Inversion operation of a dense matrix.

Solution： Efficient commute time computation with Directed graph Laplacian.

We first define the Laplacian matrix on graph as divergence of the gradient:

$$\mathbf{T}s = \mathcal{G}\mathcal{D}s, \quad \mathbf{T} = \mathbf{B}\,\mathrm{diag}\left(\{\mathbf{P}_{ij}\}_{(v_i,v_j)\in E}^{M}\right)\mathbf{B}^\top$$

Lemma 4.2. *Given a rewired graph $\widetilde{G}$, the Weighted* `DiLap` *is defined as* $\widetilde{\mathcal{T}} = \widetilde{\mathbf{\Pi}}\widetilde{\mathbf{B}}\mathrm{diag}\left(\{\widetilde{\mathbf{P}}_{ij}\}_{(v_i,v_j)\in E}^{M}\right)\widetilde{\mathbf{B}}^\top$. *Then the fundamental matrix* $\mathbf{Z}$ *of* $\widetilde{G}$ *can be solved by:*

$$\mathbf{Z} = \widetilde{\mathcal{T}}^\dagger\widetilde{\mathbf{\Pi}} = \widetilde{\mathbf{T}}^\dagger,$$

where the superscript † *means Moore–Penrose pseudoinverse of the matrix.*

Sparse matrix

Theorem 4.3. *Given $\widetilde{G}$, the hitting time and commute time from v_i to v_j on $\widetilde{G}$ can be computed as follows:*

$$h(v_i, v_j) = \frac{\widetilde{\mathbf{T}}^{\dagger}_{jj}}{\pi_j} - \frac{\widetilde{\mathbf{T}}^{\dagger}_{ij}}{\sqrt{\pi_i \pi_j}},$$

$$c(v_i, v_j) = \frac{\widetilde{\mathbf{T}}^{\dagger}_{jj}}{\pi_j} + \frac{\widetilde{\mathbf{T}}^{\dagger}_{ii}}{\pi_i} - \frac{\widetilde{\mathbf{T}}^{\dagger}_{ij}}{\sqrt{\pi_i \pi_j}} - \frac{\widetilde{\mathbf{T}}^{\dagger}_{ji}}{\sqrt{\pi_i \pi_j}}.$$

CGNN Message passing

$$m^{(\ell)}_{i,\text{in}} = \text{Agg}^{(\ell)}_{\text{in}}\left(\left\{\widetilde{\mathcal{C}}^{\text{in}}_{ij} \cdot h^{(\ell-1)}_j : v_j \in \mathcal{N}^{\text{in}}_i\right\}\right)$$

$$m^{(\ell)}_{i,\text{out}} = \text{Agg}^{(\ell)}_{\text{out}}\left(\left\{\widetilde{\mathcal{C}}^{\text{out}}_{ij} \cdot h^{(\ell-1)}_j : v_j \in \mathcal{N}^{\text{out}}_i\right\}\right)$$

$$h^{(l)}_i = \text{Comb}^{(\ell)}\left(h^{(\ell-1)}_i, m^{(\ell)}_{i,\text{in}}, m^{(\ell)}_{i,\text{out}}\right),$$

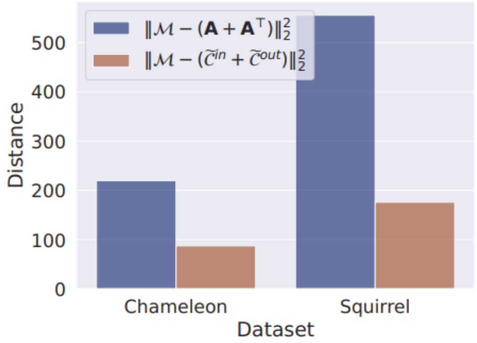
Algorithm 1 CGNN

Input: Digraph $G = (V, E, \mathbf{X})$; Depth L; Hidden size d'; Number of classes K
Output: Logits $\hat{Y} \in \mathbb{R}^{N \times K}$

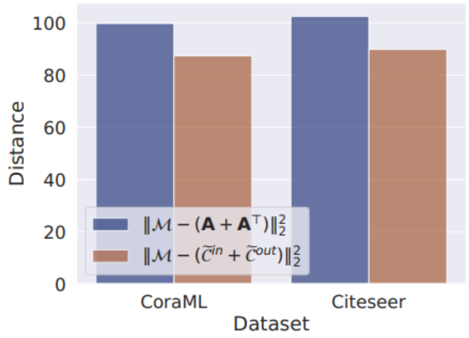
```
Compute the anchor a and node-anchor similarities to construct G' with Eq. (7).
Add all edges from G' to G to generate G̃.
Compute the Weight DiLap T̃ for G̃ with Eq. (6).
Compute R and its Moore-Penrose pseudoinverse with Eq. (8) and Eq. (27).
Compute the commute time matrix C with Eq. (10).
Compute the normalized proximity matrix C̃ with C̃^out = A ⊙ C̃ and C̃^in = A^T ⊙ C̃.
for ℓ ∈ {1, ··· , L} do
    Layer-wise message passing with Eq. (11).
end for
H = MLP(H^(L)).
Ŷ = Softmax(H).
```

Experiments

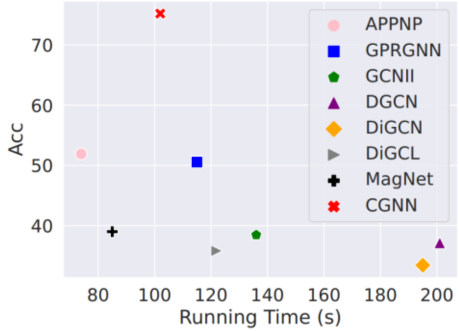
Method	Squirrel	Chameleon	Citeseer	CoraML	AM-Photo	Snap-Patents	Roman-Empire	Arxiv-Year
GCN	52.43±2.01	67.96±1.82	66.03±1.88	70.92±0.39	88.52±0.47	51.02±0.06	73.69±0.74	46.02±0.26
GAT	40.72±1.55	60.69±1.95	65.58±1.39	72.22±0.57	88.36±1.25	OOM	49.18±1.35	45.30±0.23
GraphSAGE	41.61±0.74	62.01±1.06	66.81±1.38	74.16±1.55	89.71±0.57	67.45±0.53	86.37±0.80	55.43±0.75
APPNP	51.91±0.56	45.37±1.62	66.90±1.82	70.31±0.67	87.43±0.98	51.23±0.54	72.96±0.38	50.31±0.42
MixHop	43.80±1.48	60.50±2.53	56.09±2.08	65.89±1.50	87.17±1.34	41.22±0.19	50.76±0.14	45.30±0.26
GPRGNN	50.56±1.51	66.31±2.05	61.74±1.87	73.31±1.37	90.23±0.34	40.19±0.03	64.85±0.27	45.07±0.21
GCNII	38.47±1.58	63.86±3.04	58.32±1.93	64.84±0.71	83.40±0.79	48.09±0.09	74.27±0.13	57.36±0.17
DGCN	37.16±1.72	50.77±3.31	66.37±1.93	75.02±0.50	87.74±1.02	OOM	51.92±0.43	OOM
DiGCN	33.44±2.07	50.37±4.31	64.99±1.72	77.03±0.70	88.66±0.51	OOM	52.71±0.32	48.37±0.19
MagNet	39.01±1.93	58.22±2.87	65.04±0.47	76.32±0.10	86.80±0.65	OOM	88.07±0.27	60.29±0.27
DUPLEX	57.60±0.98	61.25±0.94	67.60±0.72	72.26±0.71	87.80±0.82	66.54±0.11	79.02±0.08	64.37±0.27
DiGCL	35.82±1.73	56.45±2.77	67.42±0.14	**77.53±0.14**	89.41±0.11	70.65±0.07	87.94±0.10	63.10±0.06
DirGNN	75.19±1.26	79.11±2.28	66.57±0.74	75.33±0.32	88.09±0.46	**73.95±0.05**	91.23±0.32	64.08±0.26
CGNN	**77.83±1.52**	**79.62±2.33**	**71.59±0.16**	77.08±0.54	**90.42±0.10**	72.89±0.24	**92.87±0.45**	**66.16±0.32**



(a) Heterophilic graph.



(b) Homophilic graph.



- CGNN achieves new state-of-the-art results on 6 out of 8 datasets.
- Commute times effectively filters out irrelevant information by appropriately weighting neighbors.
- CGNN gets the best trade-off between effectiveness and efficiency.