INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS
IJCNN2025
30 JUNE - 5 JULY 2025 | ROME, ITALY
INTERNATIONAL NEURAL NETWORK SOCIETY

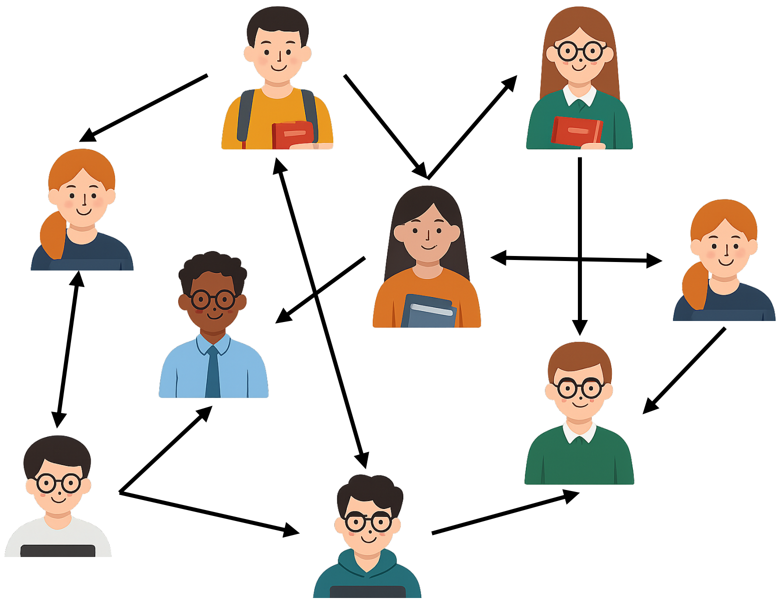
Every Hop Etched in Memory: Tokenized Graph Mamba Meets Directed Graph Learning

Lizhi Liu

China UnionPay

Directedness of Graph

- Edge directions in graphs play a unique role: reflect the flow of information.
- However, most existing graph machine learning studies focus on undirected graphs, largely ignoring directionality.



Existing Work: Extending GCN to Directed Graph

- Recent works have extended Graph Convolutional Networks (GCNs) to directed graphs by leveraging the ***complex-valued magnetic Laplacian*** to encode directionality.

$\mathbb{i} = \sqrt{-1}$ Imaginary unit

$q \in [0,0.25]$ Electric charge parameter

Magnetic Laplacian

$$\mathbf{L}^{(q)} = \mathbf{I} - \mathbf{T}^{(q)}$$

Complex Hermitian adjacency matrix

$$\mathbf{T}^{(q)} = \left(\mathbf{D}_s^{-\frac{1}{2}} \mathbf{M}_s \mathbf{D}_s^{-\frac{1}{2}}\right) \odot \exp(\mathbb{i}\boldsymbol{\Theta}^{(q)})$$

$$\mathbf{D}_s = \text{diag}(\mathbf{M}_s \mathbf{1})$$

Degree matrix

$$\mathbf{M}_s = \frac{1}{2}\left(\mathbf{M} + \mathbf{M}^{\top}\right)$$

Symmetric version of the adjacency matrix

$$\boldsymbol{\Theta}^{(q)} = 2\pi q \left(\mathbf{M} - \mathbf{M}^{\top}\right)$$

Phase matrix

Magnetic Graph Convolution

$$\mathbf{X}^{(l+1)} = \tilde{\mathbf{T}}^{(q)} \mathbf{X}^{(l)} \mathbf{W}^{(l)}$$

Self-loops added

[1] Zhang et al., NeurIPS, 2021

The Problem: Over-smoothing in Directed GCN

- **Key Issue:** Our theoretical analysis reveals that directed GCN also suffer from ***over-smoothing***, similar to their undirected counterparts.

Theorem

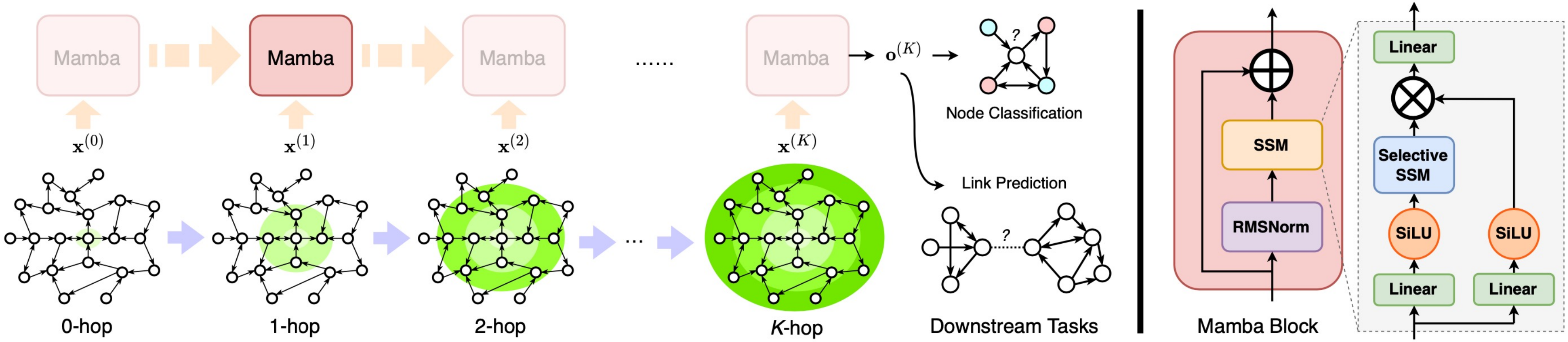
Consider the non-trivial magnetic Laplacian $\mathbf{L}^{(q)}$, assuming that $q \neq 0$ and $\mathbf{M} \neq \mathbf{M}^T$. If the directed connected graph G contains no cycles and is not bipartite, then for any $\mathbf{x} \in \mathbb{C}^N$ and $\alpha \in (0, 1]$, we have

$$\lim_{l \to +\infty} \left(\mathbf{I} - \alpha \mathbf{L}^{(q)}\right)^l \mathbf{x} = 0.$$

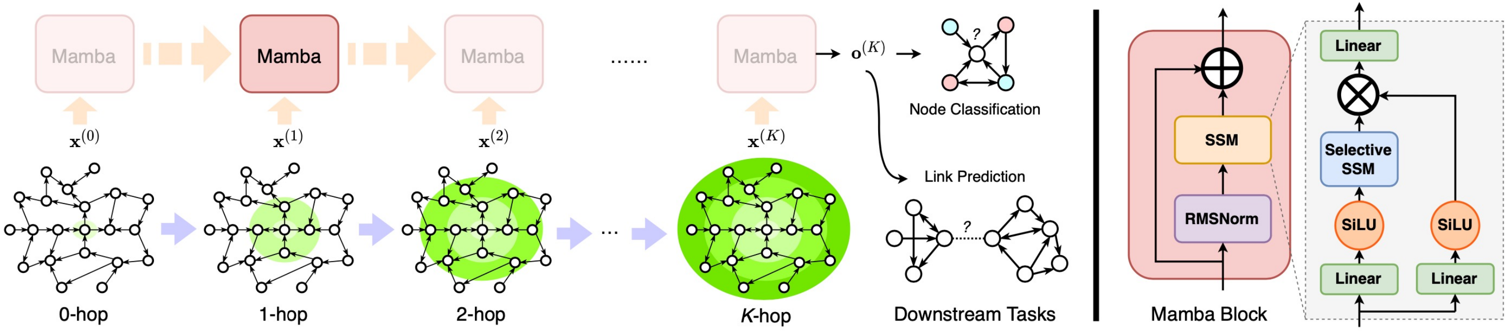
- **Implication:** As the model deepens, node signals vanish, leading to catastrophic forgetting of local information.

Proposed Method: DIGRAM

- We propose **DIGRAM**, a ***tokenized directed graph Mamba*** model, to tackle the over-smoothing problem.
- **Core Idea:** We reinterpret magnetic graph convolution's message passing as ***a token sequence generation process***.
- As new tokens emerge, information from each hop is progressively introduced into a state space model in an autoregressive fashion.



How It Works: Step-by-Step

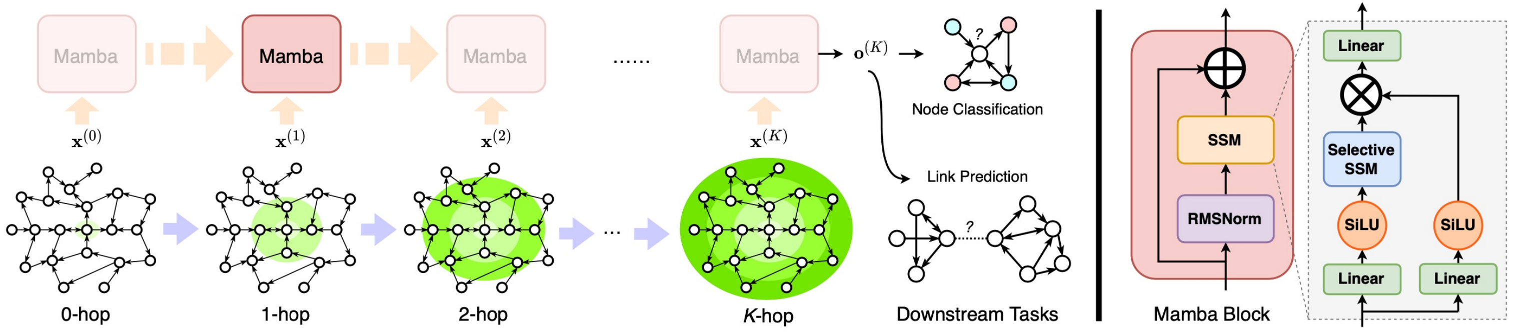


- **Token Sequence Generation**
  - We interpret the message passing of a non-parametric magnetic GCN as a generative process for a sequence of tokens.
  - The representation of the k-th token is computed via feature propagation:

$$\mathbf{X}^{(k)} = \widetilde{\mathbf{T}}^{(q)k}\mathbf{X},$$

with the initial feature defined as $\mathbf{X} = \mathbf{X}^{(0)}$.

How It Works: Step-by-Step



- **Progressive Aggregation with Mamba**
  - As new tokens (hops) are generated, we feed them into the Mamba cell autoregressively.
  - The selective state space mechanism controls information flow, allowing the model to adaptively aggregate multi-hop information.
  - It allows DIGRAM to integrate high-order topological information while storing the local context, mitigating the knowledge forgetting dilemma from over-smoothing.

Algorithm 1: The sketched procedure of DIGRAM

Input: Hermitian adjacency matrix $\tilde{\mathbf{T}}^{(q)}$; Initial features $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$; Maximum hop K

Output: Node representations $\mathbf{O}^{(K)} = \{\mathbf{o}_i^{(K)}\}_{i=1}^{N}$

Initialize $\mathbf{h}_i^{(-1)} \leftarrow 0$ for each $i = 1, \cdots, N$;

for $k \leftarrow 0$ **to** K **do**

  for $i \leftarrow 1$ **to** N **do**

    $\mathbf{z}_i^{(k)} \leftarrow \text{unwind}(\mathbf{x}_i^{(k)})$;

    $\mathbf{h}_i^{(k)} \leftarrow \tilde{\mathbf{A}}^{(k)} \odot \mathbf{h}_i^{(k-1)} + \mathbf{B}^{(k)}(\mathbf{\Delta}^{(k)} \odot \mathbf{z}_i^{(k)})$;

    $\mathbf{o}_i^{(k)} \leftarrow \mathbf{C}^{(k)}\mathbf{h}_i^{(k)} + \mathbf{D} \odot \mathbf{z}_i^{(k)}$;

  end

  $\mathbf{X}^{(k+1)} \leftarrow \tilde{\mathbf{T}}^{(q)}\mathbf{X}^{(k)}$;

end

Theoretical Justification

- The ability to express a polynomial filter with arbitrary coefficients is crucial for preventing over-smoothing.

Theorem

Given a self-looped directed graph $\tilde{G}$ and a graph signal $\mathbf{X}$, a K-layer DIGRAM model is capable of expressing a K-order polynomial frequency filter $F_K(\mathbf{X})$ with arbitrary coefficients θ_k for $k = 0, \cdots, K$.

- **Implication:** It demonstrates that DIGRAM can capture diverse graph signal patterns (low and high frequency).
- Sufficient expressiveness in the spectral domain means ***the model no longer suffers from the over-smoothing issue***.

Main Results: Node Classification

- Improvements of 1.53%, 3.02%, 2.22%, and 2.50% across four datasets.

NODE CLASSIFICATION ACCURACY (%). THE BEST RESULTS ARE IN BOLD, AND THE RUNNER-UPS ARE UNDERLINED.

	Cora-ML	CiteSeer	WikiCS	PubMed
MLP	75.79±0.70	64.10±2.07	79.28±2.03	82.30±1.30
GCN	80.97±0.74	68.33±1.92	78.30±2.01	81.62±1.32
DGCN	84.47±0.48	70.74±1.29	80.44±2.09	84.91±0.32
DiGCN	85.48±0.28	72.01±0.28	81.80±2.25	84.88±0.25
DiGCN-IB	85.64±2.02	72.24±0.75	82.56±2.30	85.11±0.31
DiGCL	75.29±2.18	62.75±1.57	69.80±2.29	75.23±0.95
MagNet	79.97±2.37	67.57±1.75	77.87±2.11	84.69±0.75
HoloNets	<u>85.78±1.83</u>	72.55±0.92	80.91±1.42	84.23±1.97
DiGAE	81.14±0.47	69.38±2.06	78.26±1.89	81.59±1.24
Dir-GNN	85.30±0.57	71.79±2.22	82.27±1.60	83.36±0.52
LightDiC	78.96±1.45	66.15±0.92	79.75±0.99	69.68±1.06
DUPLEX	85.31±1.78	<u>72.85±1.27</u>	<u>83.04±1.31</u>	<u>85.63±0.23</u>
DIGRAM	**87.31±0.36**	**75.87±0.53**	**85.26±0.42**	**88.13±0.29**

Main Results: Link Prediction

- DIGRAM achieves state-of-the-art results in all 12 cases.

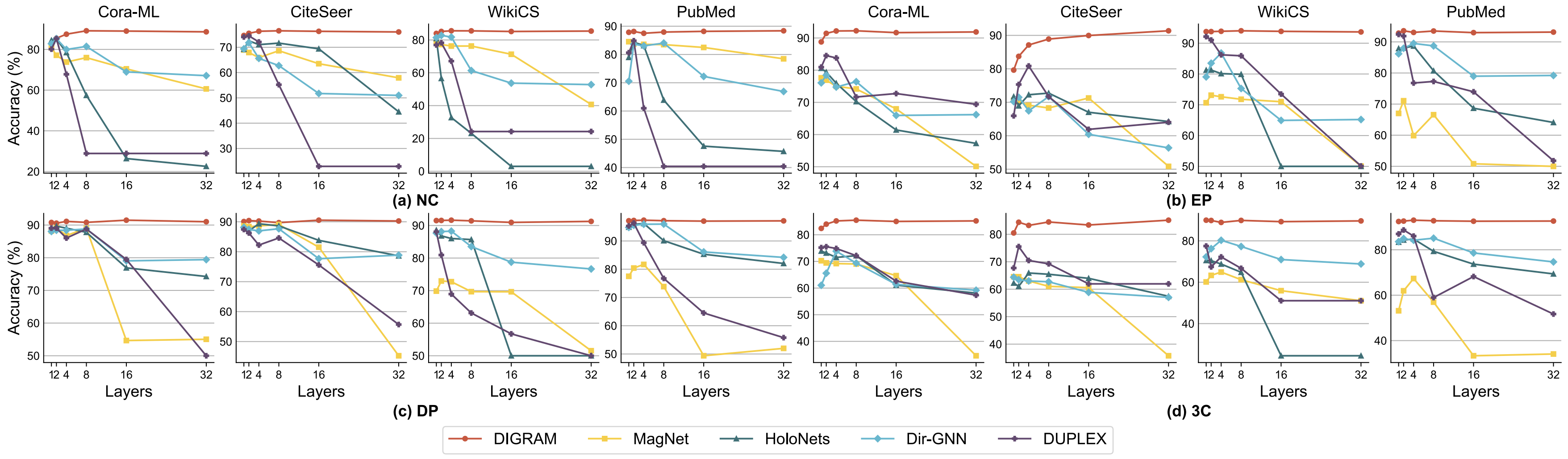
LINK PREDICTION ACCURACY (%). THE BEST RESULTS ARE IN BOLD, AND THE RUNNER-UPS ARE UNDERLINED.

	EP				DP				3C			
	Cora-ML	CiteSeer	WikiCS	PubMed	Cora-ML	CiteSeer	WikiCS	PubMed	Cora-ML	CiteSeer	WikiCS	PubMed
MLP	78.81±0.85	68.51±1.98	87.53±0.79	87.82±2.22	89.04±1.02	88.86±1.20	86.55±2.24	94.92±0.66	71.66±0.46	65.14±1.47	77.23±1.56	84.19±1.54
GCN	77.38±0.47	69.57±0.44	85.02±2.30	76.35±0.97	85.89±2.06	81.52±0.62	79.93±0.98	83.19±1.97	72.73±1.47	63.17±1.41	71.32±1.96	63.82±1.62
DGCN	76.79±0.79	65.74±1.34	86.52±1.71	89.51±1.94	88.92±1.40	87.91±0.32	86.23±0.87	95.56±1.81	70.84±0.57	66.82±1.16	80.22±0.72	84.24±0.98
DiGCN	72.98±1.34	67.87±1.55	82.88±1.08	84.18±1.16	88.16±1.04	87.68±1.19	83.36±1.94	94.74±1.00	68.37±0.36	62.56±2.16	73.89±0.64	80.60±1.31
DiGCN-IB	74.76±1.19	71.28±0.77	81.70±0.67	88.54±0.89	90.43±0.40	89.10±1.12	84.16±1.50	95.62±2.10	71.33±1.75	63.17±2.08	74.92±0.84	83.42±2.10
DiGCL	64.88±0.92	59.87±0.89	76.32±1.50	71.39±1.93	72.29±1.63	68.72±1.33	69.64±1.28	81.02±1.57	46.79±2.22	38.81±1.89	61.10±1.01	56.89±1.68
MagNet	77.50±2.11	68.30±0.76	72.08±1.45	71.07±0.77	90.43±1.07	88.39±1.60	72.79±0.61	81.66±1.08	70.59±1.93	64.69±2.04	64.91±0.85	67.34±0.36
HoloNets	80.00±0.78	71.49±0.83	80.71±1.10	88.52±0.80	89.04±1.79	88.39±1.03	86.01±2.05	96.14±0.77	71.99±2.07	64.23±1.59	68.49±0.40	84.31±2.29
DiGAE	65.60±1.25	61.91±1.92	73.95±2.14	60.31±0.69	71.16±2.13	56.40±1.96	59.83±0.76	55.19±0.50	51.32±1.17	47.79±0.83	56.87±1.77	41.76±1.54
Dir-GNN	79.29±0.32	69.36±1.41	86.87±1.82	90.12±2.04	89.80±0.57	88.39±1.72	88.67±0.64	95.78±2.01	74.88±1.78	64.54±0.91	80.50±0.57	85.05±1.68
LightDiC	72.38±1.33	65.53±1.45	83.73±0.32	77.93±0.45	75.81±0.75	83.89±0.72	85.38±1.39	80.08±1.88	64.61±0.93	64.99±0.86	78.73±0.36	70.55±2.25
DUPLEX	81.31±1.14	80.85±1.85	90.66±0.39	91.36±1.33	88.04±0.91	87.44±1.53	88.07±0.51	96.32±0.65	74.22±1.15	76.41±0.67	77.50±1.45	86.70±0.53
DIGRAM	**92.12±0.45**	**86.85±0.28**	**93.69±0.57**	**92.63±0.15**	**91.00±0.20**	**89.71±0.46**	**91.64±0.18**	**97.32±0.17**	**84.75±0.28**	**83.34±0.50**	**89.25±0.69**	**93.04±0.22**

Three subtasks: Existence Prediction (EP), Direction Prediction (DP), Three-class Prediction (3C)

Key Finding: Mitigating Over-smoothing

- DIGRAM is the only method that maintains stable or even improves performance as layers increase on both node classification (NC) and link prediction (EP, DP, 3C) tasks.
- In contrast, SOTA methods experience a sharp performance decline as the network deepens.
- It strongly supports our claim that tokenized graph Mamba effectively mitigates the over-smoothing problem.



Conclusion, Limitations & Future Work

- **Summary**
  - We introduce **DIGRAM**, a tokenized graph Mamba model for directed graph learning.
  - By treating message passing as a token sequence generation process, DIGRAM simultaneously captures local and global topological contexts.
  - DIGRAM achieves superior performance on node classification and link prediction tasks across 16 cases compared with SOTA methods.
  - Crucially, it demonstrates robustness against the over-smoothing problem as model depth increases.

- **Limitations**
  - **Parameter Sensitivity:** The performance of the model is somewhat sensitive to the choice of the charge parameter q, requiring careful manual tuning.
  - **Heterophily Challenge**: Similar to other models based on the classical message-passing framework, its performance on heterophilic directed graphs remains suboptimal.

- **Future Work**
  - Exploring strategies to address heterophily mixing.
  - Extending the methodology to signed graphs.

Thank you!

GitHub

Homepage