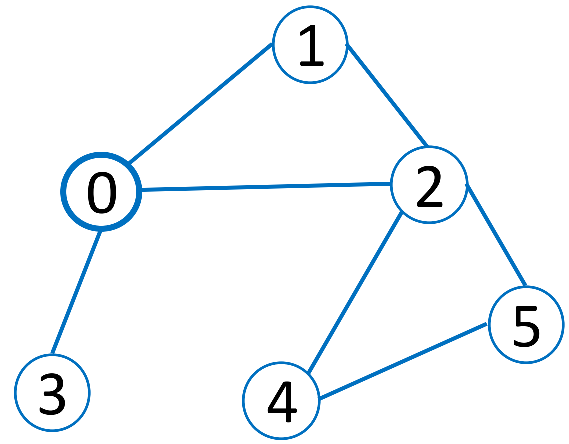
TGB-Seq Benchmark: Challenging Temporal GNNs with Complex Sequential Dynamics

Lu Yi, Jie Peng, Yanping Zheng*, Fengran Mo, Zhewei Wei*
Yuhang Ye, Zixuan Yue, Zengfeng Huang

Contact: yilu@ruc.edu.cn

Temporal Graphs

- Static graphs $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$
  - The graph remains unchanged over time.



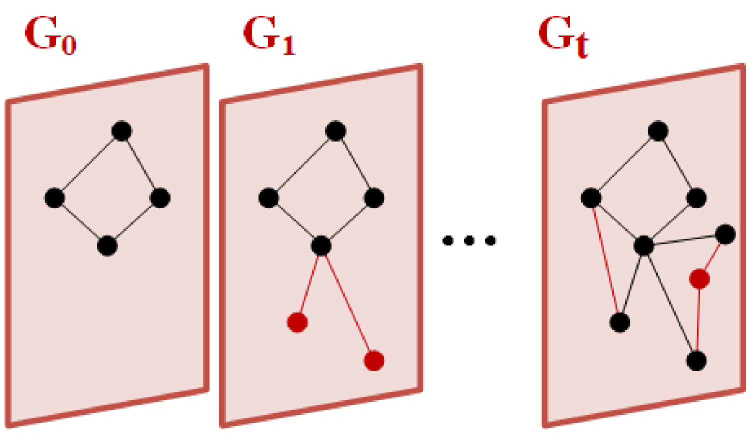
- Temporal graphs: a more concrete abstraction of real-world systems
  - Social network: new users join, connections between users form
  - Recommendation network: users purchase and review products
  - $G_i = (V_i, E_i)$

Temporal Graphs

<u>Discrete-time temporal graphs</u>

- A stream of graph snapshots:

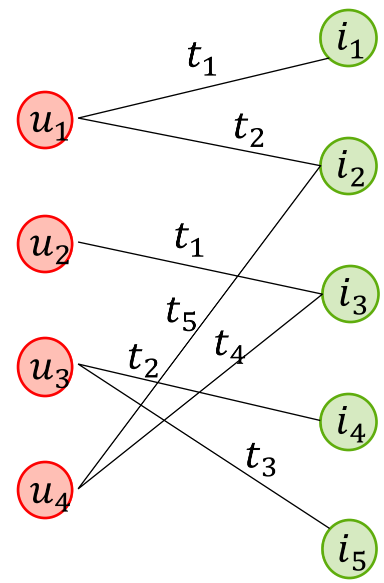
$$\{G_1, G_2, \ldots, G_T\}$$
$$G_i = (V_i, E_i)$$



<u>**Continuous-time temporal graphs**</u>

- A stream of edges:

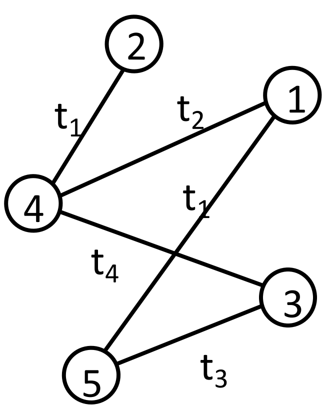
$$\{(s_0, d_0, t_0), (s_1, d_1, t_1), \cdots, (s_T, d_T, t_T)\}$$



Tasks of Temporal GNNs

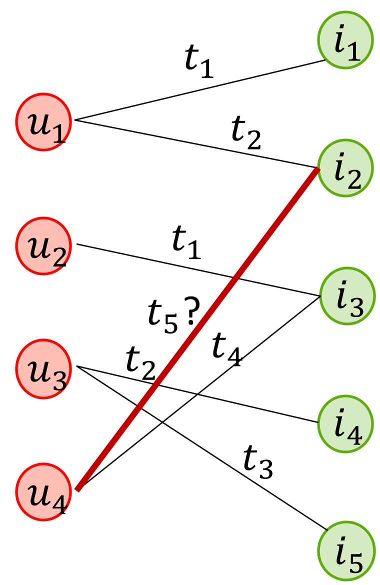
- **Future link prediction**
  - Given G, (s, d, t) and E before t, the model is asked to predict the likelihood of the edge (s, d) appearing at time t.
  - Often framed as a **ranking** problem among multiple negative samples. Given G, (s, d, t) and E before t, the model is asked to rank d higher among sampled k nodes.
  - MRR is used as the metric.

- Other tasks: dynamic node classification…



$\longrightarrow \quad y(v_1 | t_5) = ?$

dynamic node classification



future link prediction

Temporal GNNs

- Existing methods = memory module + aggregation module
- Memory module

$$\mathbf{mem}(s) = \text{MEM}\left(\mathbf{mem}(s), \mathbf{x}_s, \{(\mathbf{mem}(d), \mathbf{e}_{s,d}, \Delta t) \mid d \in \mathcal{N}_{\mathbf{b}}(s)\}\right),$$

- Aggregation module

$$\mathbf{emb}(s) = \text{AGGR}\left(\mathbf{x}_s, \left\{(\mathbf{x}_d, \mathbf{e}_{s,d}, \Delta t) \mid d \in \mathcal{N}_t^k(s)\right\}\right),$$

  - If memory module is available, x_s is the combination of the memory and feature.

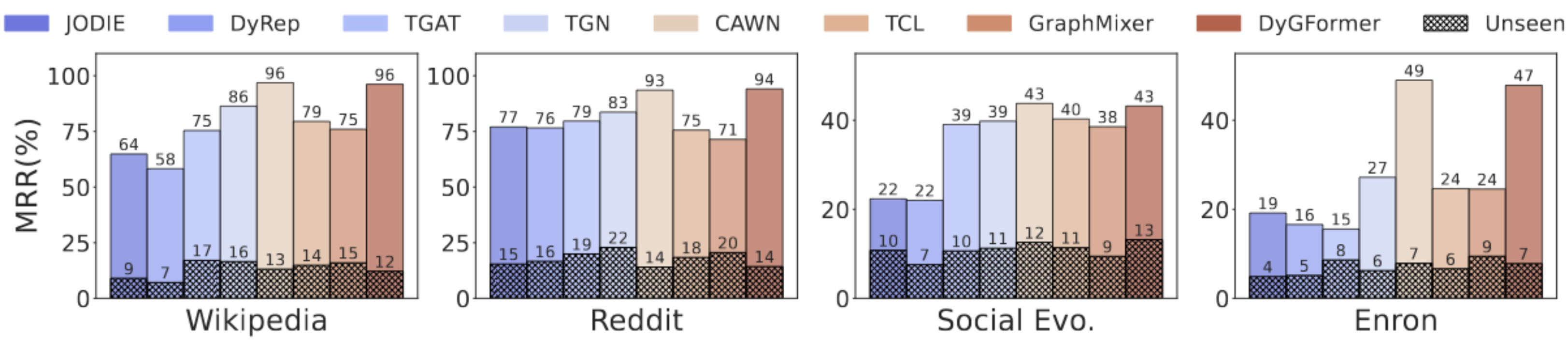
- High-order aggregation module

$$\mathbf{rel}(s, d) = \text{CO-REL}\left(\mathcal{N}_t^k(s), \mathcal{N}_t^k(d)\right),$$

Method	Memory	Aggregation	High-order
TGN	GRU	Attention	/
EdgeBank	Record all histories	/	/
GraphMixer	/	MLP-Mixer	/
DyGFormer	/	Attention	Co-neighbors frequency

Motivations

- Observation 1. Temporal GNNs excel in predicting seen edges but struggle to generalize to unseen edges.



- Observation 2. Temporal GNNs fail to perform effectively on recommendation datasets, a typical downstream application.

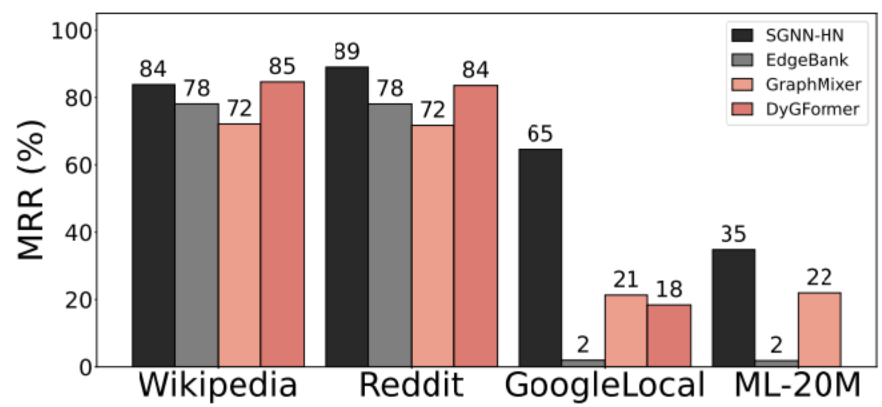
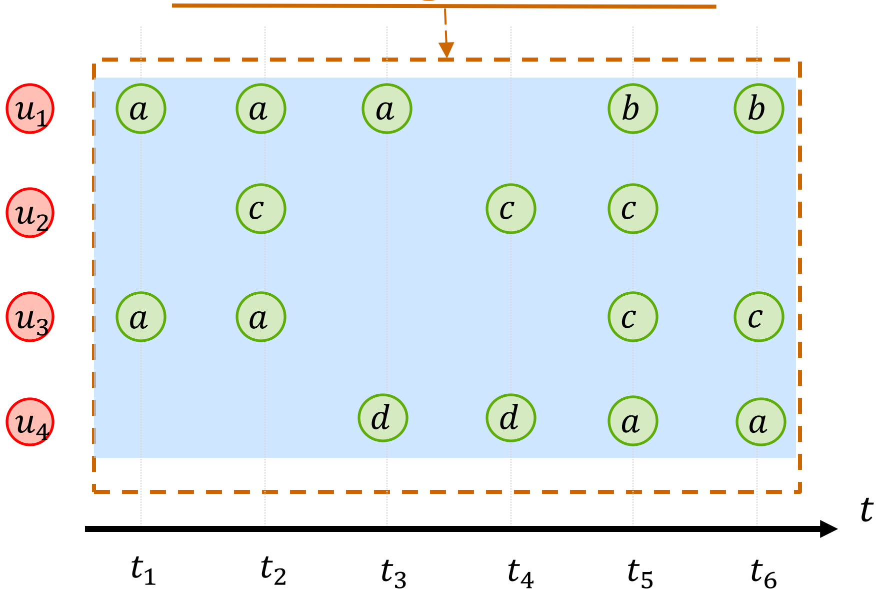


Figure 1: The MRR scores of three selected temporal GNNs and SGNN-HN on two existing datasets (Wikipedia, Reddit) and two recommendation datasets (Yelp and Taobao).

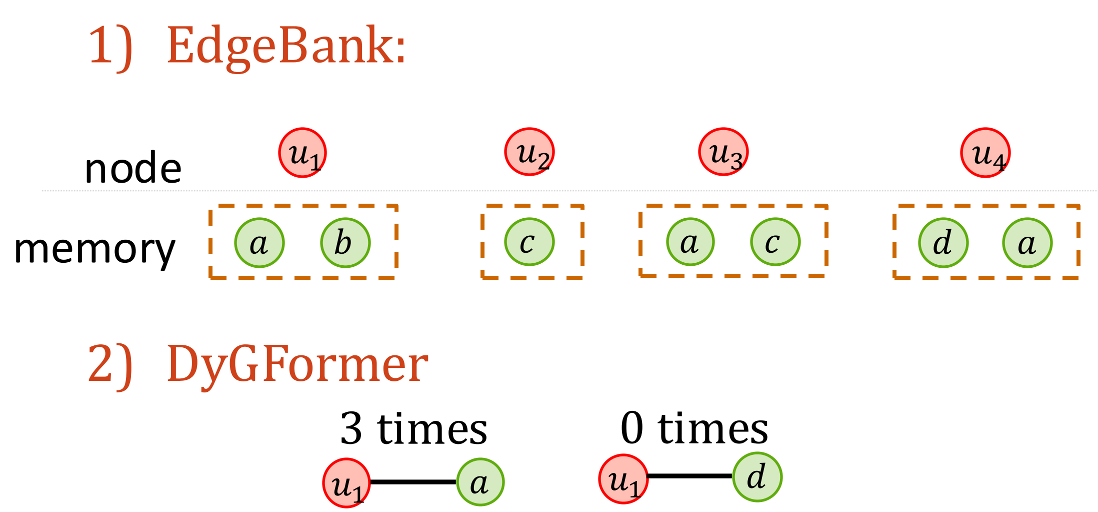
Existing dataset

- Existing datasets (e.g., Wikipedia, Reddit) often contain excessive repeated historical edges.
- Leading existing methods to predict historical edges using memory or aggregation techniques.
- These methods perform well on such datasets but may struggle to generalize.

Existing datasets



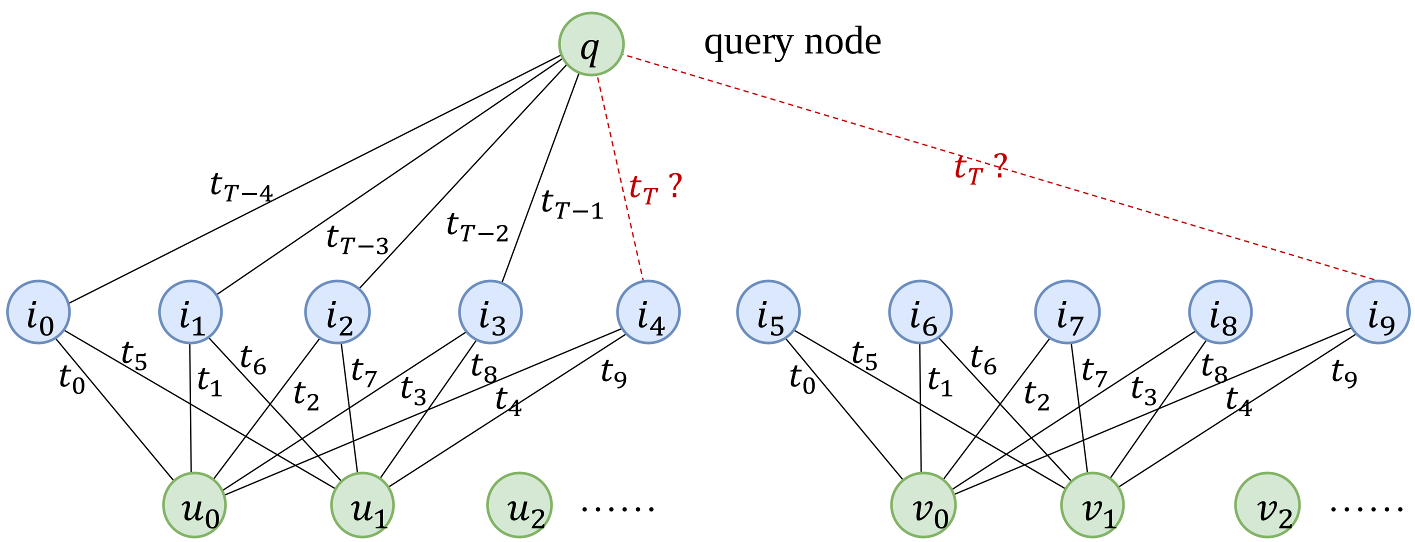
Existing temporal GNNs



u_1 is more likely to interact with a instead of d.

Current pitfalls in temporal GNNs

- Existing temporal GNNs struggle to capture even basic sequential dynamics, limiting their effectiveness in real-world applications



When predicting the next interaction of a node which has previously interacted with i_0, i_1, i_2, i_3, all methods fail to correctly predict i_4 instead of i_9.

Table 1: The AP metric on the toy example dataset. ℓ indicates the length of the temporal random walk of CAWN.

Method	AP (%)
JODIE	51.19 ± 0.32
DyRep	51.30 ± 0.27
TGAT	51.06 ± 0.23
TGN	51.25 ± 0.48
CAWN ($\ell = 1$)	50.00 ± 0.00
CAWN ($\ell = 2$)	52.80 ± 0.05
EdgeBank	50.00 ± 0.00
TCL	50.00 ± 0.00
GraphMixer	50.00 ± 0.00
DyGFormer	50.66 ± 0.50
SGNN-HN	100.00 ± 0.00

Analysis of the Pitfall

- Memory module

$$\mathbf{mem}(s) = \text{MEM}\left(\mathbf{mem}(s), \mathbf{x}_s, \{(\mathbf{mem}(d), \mathbf{e}_{s,d}, \Delta t) \mid d \in \mathcal{N}_{\mathbf{b}}(s)\}\right),$$

- Aggregation module

$$\mathbf{emb}(s) = \text{AGGR}\left(\mathbf{x}_s, \left\{(\mathbf{x}_d, \mathbf{e}_{s,d}, \Delta t) \mid d \in \mathcal{N}_t^k(s)\right\}\right),$$

- High-order aggregation module

$$\mathbf{rel}(s, d) = \text{CO-REL}\left(\mathcal{N}_t^k(s), \mathcal{N}_t^k(d)\right),$$

- Due to the lack of distinguishing feature in nodes and edges, and the identical interaction times between group u and group v, both modules cannot distinguish i_4 and i_9 in the toy example.
- High-order aggregation offers slight improvement
  - but introduces excessive noise [Besta et al. 2024]

Method	AP (%)
CAWN ($\ell = 1$)	50.00 ± 0.00
CAWN ($\ell = 2$)	52.80 ± 0.05

Sequential dynamics are everywhere!

Movie watching

Social networks

E-commerce

- <u>Understanding **complex** sequential dynamics is key to improving temporal GNNs performance in real-world applications.</u>

TGB-Seq: new benchmark

- focuses on the task of future link prediction
- eight datasets: four bipartite and four non-bipartite graphs
- low repeat ratio $r = |\mathcal{E}_{\text{seen}}|/|\mathcal{E}|$
- diverse domains representing typical real-world application of future link prediction
- adhere to power-law degree distributions
- medium to large scale

Table 2: Statistics of TGB-Seq datasets.

Dataset	Nodes (users/items)	Edges	Timestamps	Repeat ratio(%)	Density(%)	Bipartite	Domain
ML-20M	100,785/9,646	14,494,325	9,993,250	0	1.49×10^{0}	✓	Movie rating
Taobao	760,617/863,016	18,853,792	139,171	16.58	2.87×10^{-3}	✓	E-commerce interaction
Yelp	1,338,688/405,081	19,760,293	14,646,734	25.18	3.64×10^{-3}	✓	Business review
GoogleLocal	206,244/267,336	1,913,967	1,771,060	0	3.47×10^{-3}	✓	Business review
Flickr	233,836	7,223,559	134	0	1.32×10^{-2}	×	Who-To-Follow
YouTube	402,422	3,288,028	203	0	2.03×10^{-3}	×	Who-To-Follow
Patent	2,241,784	12,749,824	1,632	0	2.54×10^{-4}	×	Citation
WikiLink	1,361,972	34,163,774	2,198	0	1.84×10^{-3}	×	Web link

Compare with existing datasets

- low repeat ratio $r = |\mathcal{E}_{\text{seen}}|/|\mathcal{E}|$

Table 5: A selected list of datasets used for continuous-time temporal graph learning.

Dataset	Nodes (users/items)	Edges	Timestamps	Repeat ratio(%)	Density(%)	Bipartite	Domain
ML-20M	100,785/9,646	14,494,325	9,993,250	0	1.49×10^{0}	✓	Movie rating
Taobao	760,617/863,016	18,853,792	139,171	16.58	2.87×10^{-3}	✓	E-commerce interaction
Yelp	1,338,688/405,081	19,760,293	14,646,734	25.18	3.64×10^{-3}	✓	Business review
GoogleLocal	206,244/267,336	1,913,967	1,771,060	0	3.47×10^{-3}	✓	Business review
Flickr	233,836	7,223,559	134	0	1.32×10^{-2}	×	Who-To-Follow
YouTube	402,422	3,288,028	203	0	2.03×10^{-3}	×	Who-To-Follow
Patent	2,241,784	12,749,824	1,632	0	2.54×10^{-4}	×	Citation
WikiLink	1,361,972	34,163,774	2,198	0	1.84×10^{-3}	×	Web link
Wikipedia	8,227/1,000	157,474	152,757	88.41	1.91×10^{0}	✓	Interaction
Reddit	10,000/984	672,447	669,065	88.32	6.83×10^{0}	✓	Social
MOOC	7,047/97	411,749	345,600	56.66	6.02×10^{1}	✓	Interaction
LastFM	980/1,000	1,293,103	1,283,614	88.01	1.32×10^{2}	✓	Interaction
Enron	184	125,235	22,632	90.79	3.70×10^{2}	×	Social
Social Evo.	74	2,099,519	565,932	99.77	3.83×10^{4}	×	Proximity
UCI	1,899	59,835	58,911	66.06	1.66×10^{0}	×	Social
Flights	13,169	1,927,145	122	79.50	1.11×10^{0}	×	Transport
Contact	692	2,426,279	8,064	96.72	5.07×10^{2}	×	Proximity
tgbl-wiki	8,227/1,000	157,474	152,757	88.41	1.91×10^{0}	✓	Interaction
tgbl-review	352,636/298,590	4,873,540	6,865	0.19	4.63×10^{-3}	✓	Rating
tgbl-coin	638,486	22,809,486	1,295,720	82.93	5.60×10^{-3}	×	Transaction
tgbl-comment	994,790	44,314,507	30,998,030	19.81	4.48×10^{-3}	×	Social
tgbl-flight	18,143	67,169,570	1,385	96.48	2.04×10^{1}	×	Transport
Bitcoin-Alpha	3,783	24,186	24,186	0	1.69×10^{-1}	×	Finance
Bitcoin-OTC	5,881	35,592	35,592	0	1.03×10^{-1}	×	Finance

Node Degree Distribution

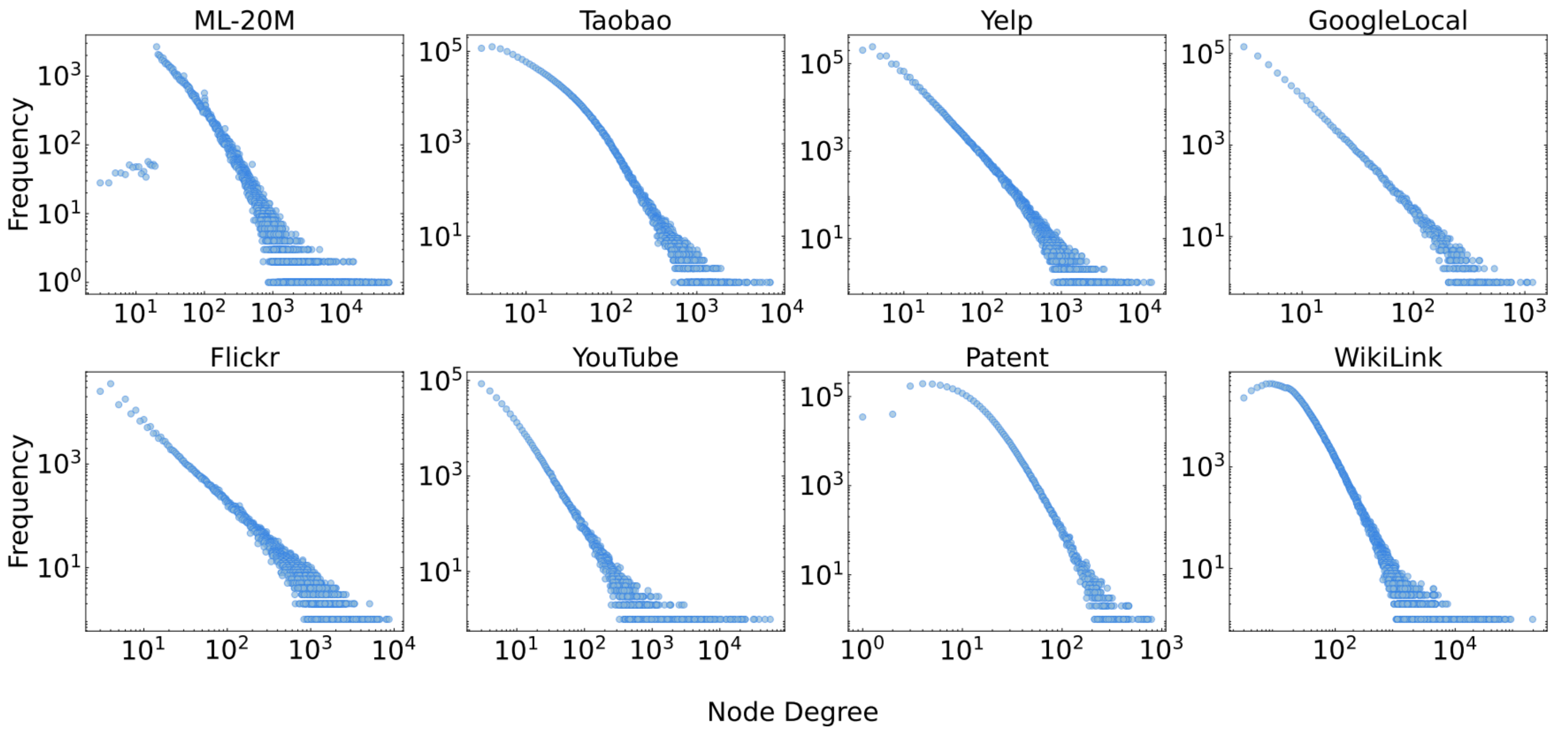


Figure 5: Distribution of node degree on our TGB-Seq dataset.

Benchmarking (1)

Datasets	ML-20M	Taobao	Yelp	GoogleLocal	Wikipedia	Reddit
JODIE	21.16 ± 0.73	55.25 ± 0.35	69.88 ± 0.31	41.86 ± 1.49	76.94 ± 0.28	77.92 ± 0.10
DyRep	19.00 ± 1.69	49.67 ± 1.41	57.69 ± 1.05	37.73 ± 1.34	68.09 ± 1.45	75.30 ± 0.30
TGAT	10.47 ± 0.20	OOT	OOT	19.78 ± 0.24	72.42 ± 0.38	76.69 ± 0.52
TGN	23.99 ± 0.20	62.21 ± 4.09	69.79 ± 0.24	54.13 ± 1.97	81.16 ± 0.19	79.82 ± 0.26
CAWN	12.31 ± 0.02	OOT	25.71 ± 0.09	18.26 ± 0.02	88.23 ± 0.33	87.31 ± 0.32
EdgeBank	1.82 ± 0.00	OOT	9.77 ± 0.00	1.96 ± 0.00	78.10 ± 0.00	78.08 ± 0.00
TCL	12.04 ± 0.02	31.55 ± 0.03	24.39 ± 0.67	18.30 ± 0.02	45.47 ± 3.48	36.09 ± 2.10
GraphMixer	21.97 ± 0.17	31.54 ± 0.02	33.96 ± 0.19	21.31 ± 0.14	72.14 ± 0.80	71.73 ± 0.32
DyGFormer	OOT	OOT	21.68 ± 0.20	18.39 ± 0.02	84.64 ± 0.43	83.57 ± 1.42
SGNN-HN	33.12 ± 0.01	68.58 ± 0.21	69.34 ± 0.44	62.88 ± 0.51	83.83 ± 0.55	89.01 ± 0.17

- Memory-based methods (JODIE, DyRep and TGN) significantly outperform other temporal GNNs on TGB-Seq datasets, in contrast with their performance on the Wikipedia and Reddit dataset.
- DyGFormer and CAWN outperform other temporal GNNs on Wikipedia and Reddit dataset, but fail to perform effectively on TGB-Seq datasets.
- TGB-Seq datasets assess the capabilities of temporal GNNs from a novel perspective, distinct from existing datasets.

Benchmarking (2)

Datasets	Flickr	YouTube	Patent	WikiLink
JODIE	60.43 ± 1.63	65.21 ± 0.50	20.80 ± 1.00	70.43 ± 1.26
DyRep	60.28 ± 1.41	63.02 ± 1.92	22.67 ± 0.27	60.25 ± 1.68
TGAT	23.53 ± 3.35	43.56 ± 2.53	8.49 ± 0.18	OOT
TGN	68.38 ± 0.57	72.06 ± 1.48	20.92 ± 1.22	73.84 ± 2.96
CAWN	48.69 ± 6.08	47.55 ± 1.08	12.34 ± 0.47	OOT
TCL	40.00 ± 1.76	50.17 ± 1.98	10.60 ± 1.75	43.02 ± 2.16
GraphMixer	45.01 ± 0.08	58.87 ± 0.12	18.97 ± 2.54	48.57 ± 0.02
DyGFormer	49.58 ± 2.87	46.08 ± 3.44	14.20 ± 2.93	OOT

- The memory-based methods outperform others across datasets.
- Aggregation-only methods perform better on non-bipartite datasets than on bipartite datasets.
- The rank of aggregation-only methods are various across datasets: DyGFormer > GraphMixer on Flickr, GraphMixer > DyGFormer on YouTube.

Training cost

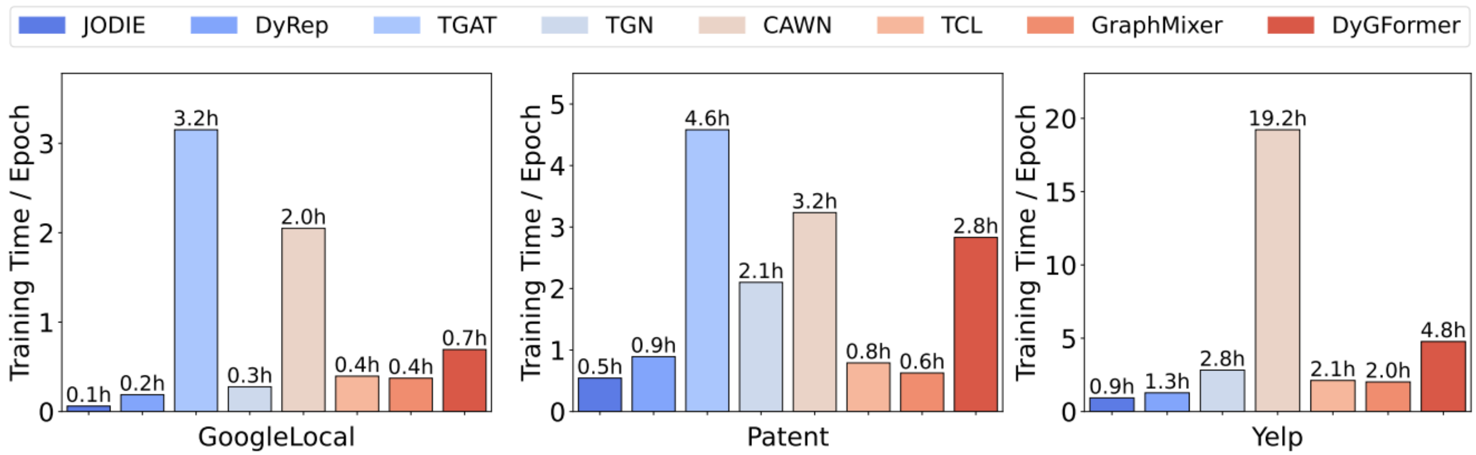


Figure 4: The average training cost per epoch of nine popular temporal GNN methods on GoogleLocal, Patent, and Yelp datasets consists of 1.9M, 12.7M, and 19.7M edges, respectively.

- Most efficient methods: JODIE, DyRep, GraphMixer, TCL, TGN
- Inefficient methods: TGAT, DyGFormer, CAWN

Conclusion

- Existing temporal GNNs fail to capture sequential dynamics in temporal graphs, limiting their generalizations to unseen edges.
- Existing datasets contain excessive repetitions of edges and overlook the intricate sequential dynamics present in real-world dynamic systems.
- TGB-Seq datasets are curated from diverse application domains with intricate sequential dynamics and minimal repeated edges.
- Benchmarking on TGB-Seq datasets highlights the limitations of existing temporal GNNs, demonstrates TGB-Seq's ability to evaluate temporal GNNs from a distinct perspective compared to existing datasets.

Resources

TGB-SEQ

Temporal Graph Benchmark with Sequential Dynamics

- pip package: `pip install tgb-seq`.
- Website: https://tgb-seq.github.io/, including TGB-Seq leaderboard and documentation.
- GitHub: https://github.com/TGB-Seq/TGB-Seq.

Thank you!

The ALGO Lab

-人工智能与大数据基础算法实验室-