Temporal Graph Benchmark for Machine Learning on Temporal Graphs

Temporal Graph Benchmark

[Website](), [Paper](), [Github](), [Pypi](), [Documentation]()
https://tgb.complexdatalab.com/

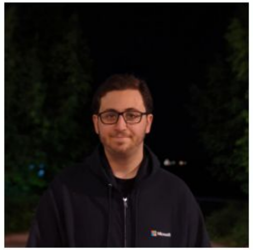
NeurIPS 2023 Datasets and Benchmarks Track

Presented by Shenyang Huang

TGB team

Shenyang Huang
Mila, McGill University

Farimah Poursafaei
Mila, McGill University



Jacob Danovitch
Mila, McGill University

Matthias Fey
kumo.ai

Weihua Hu
kumo.ai

Emanuele Rossi
Imperial College London

Jure Leskovec
Stanford University

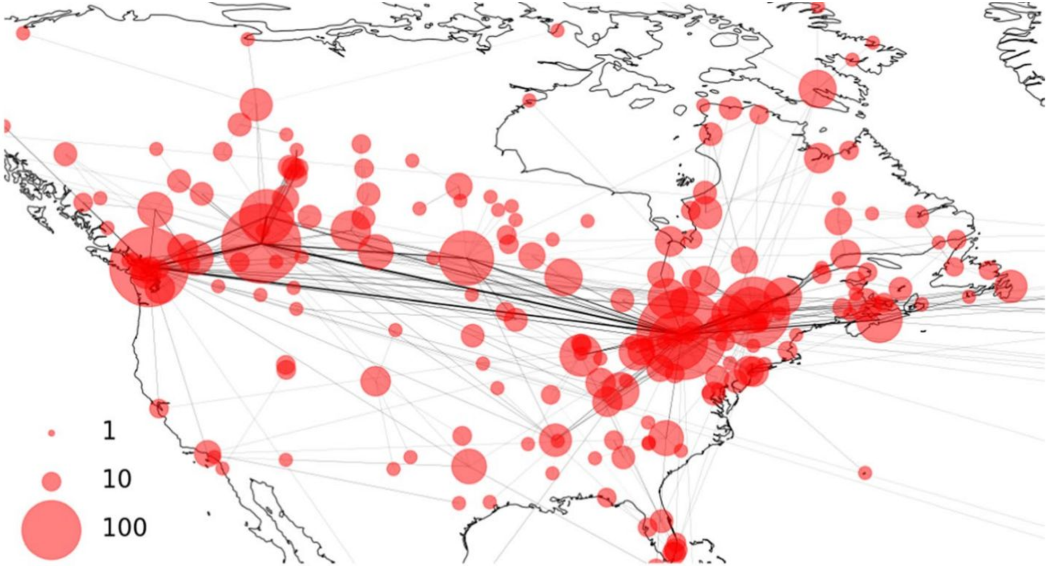
Michael Bronstein
University of Oxford

Guillaume Rabusseau
Mila, Université de Montréal, CIFAR AI Chair

Reihaneh Rabbany
Mila, McGill University, CIFAR AI Chair

Many Networks are Temporal

- Social Networks
- Traffic Networks
- Financial Networks
- Political Networks
- Interaction Networks
- More



Canadian Flight Network on April 2nd, 2020

Incorporating dynamic fight network in SEIR to model mobility between populations

Continuous Time Dynamic Graphs

Timestamped edge streams $G = \{(s_0,d_0,t_0), (s_1,d_1,t_1), \ldots, (s_T,d_T,t_T)\}$

With node features X_t and edge features M_t

Streaming setting: previously observed test edges can be accessed by the model but back-propagation and weight updates with the test information are not permitted.

Motivations for TGB

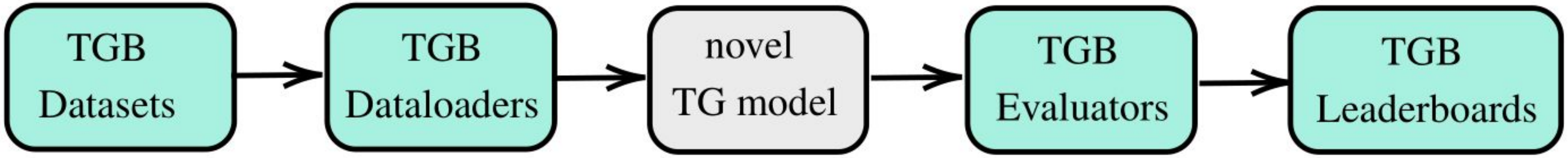
1. Lack of realistic and large scale datasets.

2. Lack of standardized evaluation similar to OGB for TG.

3. High performance of methods hinders the ability to differentiate methods.

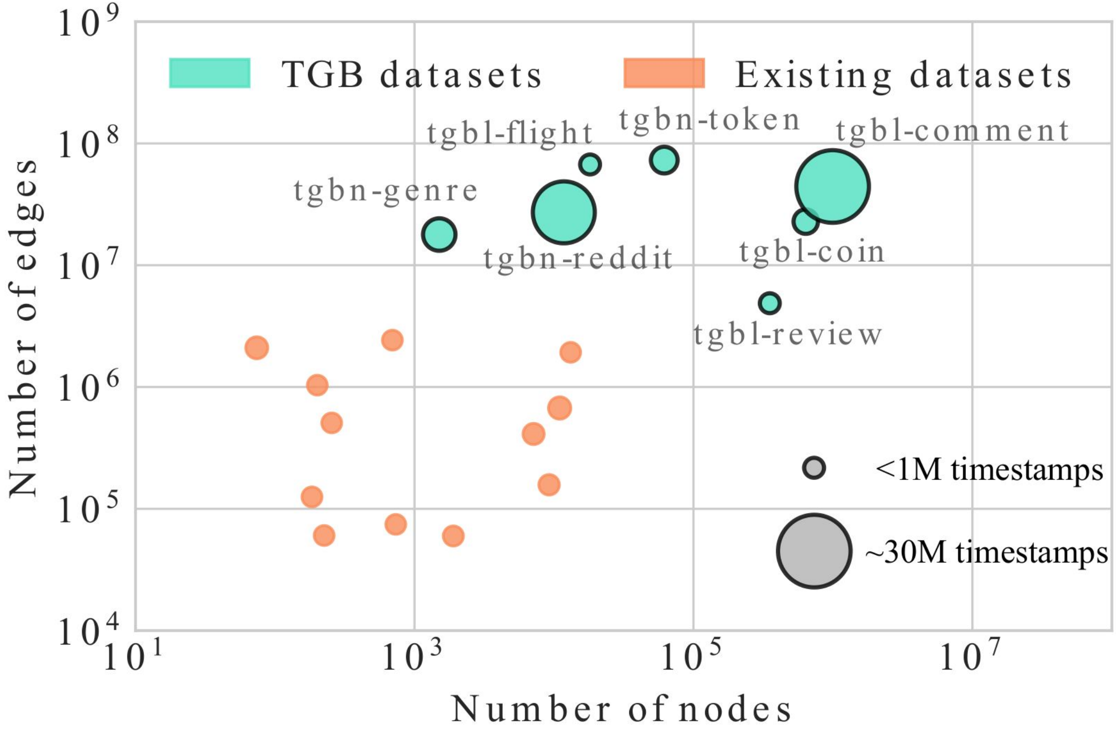
Poursafaei et al. Towards Better Evaluation for Dynamic Link Prediction, 2022

TGB
Temporal Graph Benchmark



- **Large-scale** and **realistic** datasets from **five domains**
- Both dynamic **link** and **node** property prediction task
- TGB package **automatically downloads datasets** and processes them in `numpy`, `PyTorch` and `PyG`
- **Reproducible and Realistic** Evaluation Protocol
- Public and **Online Leaderboard** to track recent developments

TGB Datasets



➢ Orders of magnitude larger than standard benchmark datasets in **nodes, edges and timestamps**

Dynamic Link Property Prediction

➢ Predict property (often existence) of a link at a future timestamp

Link Prediction Evaluation for an edge (u,v,t)

- sample k negative edges with the same source u, at time t
- sample half historical edges and half random negatives
- evaluated by Mean Reciprocal Rank (MRR)

a sampled set of negative edges are provided for each dataset for reproducibility

Surprise Index & Method Ranking

Table 2: Results for *dynamic link property prediction* on small datasets.

Method	MRR	
	Validation	Test
DyRep [42]	0.072±0.009	0.050±0.017
TGN [35]	0.435±0.069	0.396±0.060
CAWN [46]	<u>0.743</u>±0.004	<u>0.711</u>± 0.006
TCL [45]	0.198±0.016	0.207±0.025
GraphMixer [9]	0.113±0.003	0.118±0.002
TGAT [48]	0.131±0.008	0.141±0.007
NAT [25]	**0.773**±0.011	**0.749**±0.010
$\text{EdgeBank}_{\text{tw}}$ [33]	0.600	0.571
EdgeBank_{∞} [33]	0.527	0.495

tgbl-wiki *Surprise index* = 0.108

Method	MRR	
	Validation	Test
DyRep [42]	0.216±0.031	0.220±0.030
TGN [35]	0.313±0.012	0.349±0.020
CAWN[46]	0.200±0.001	0.193±0.001
TCL [45]	0.199±0.007	0.193±0.009
GraphMixer [9]	**0.428**±0.019	**0.521**±0.015
TGAT [48]	<u>0.324</u>±0.006	<u>0.355</u>±0.012
NAT [25]	0.302±0.011	0.341±0.020
$\text{EdgeBank}_{\text{tw}}$ [33]	0.0242	0.0253
EdgeBank_{∞} [33]	0.0229	0.0229

tgbl-review *Surprise index* = 0.987

$$\textit{Surprise index} = \frac{|E_{test} \setminus E_{train}|}{|E_{test}|}$$

➢ Method rankings change significantly with different surprise index

Medium and Large Datasets

Method	tgbl-coin MRR Validation	tgbl-coin MRR Test	tgbl-comment MRR Validation	tgbl-comment MRR Test	tgbl-flight MRR Validation	tgbl-flight MRR Test
DyRep [42]	0.512±0.014	0.452±0.046	0.291±0.028	0.289±0.033	0.573±0.013	0.556±0.014
TGN [35]	**0.607**±0.014	**0.586**±0.037	**0.356**±0.019	**0.379**±0.021	**0.731**±0.010	**0.705**±0.020
$\text{EdgeBank}_{\text{tw}}$ [33]	0.492	0.580	0.124	0.149	0.363	0.387
EdgeBank_{∞} [33]	0.315	0.359	0.109	0.129	0.166	0.167
	Medium scale		Large scale		Large scale	

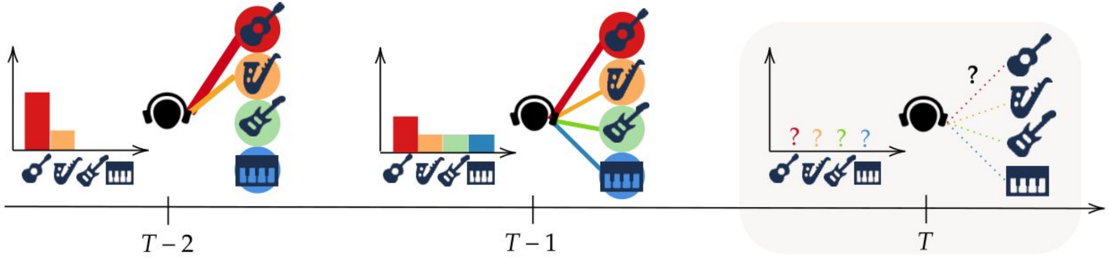
- Many TG methods are too expensive to run on the larger datasets can take weeks or more
- Develop scalable methods is an important direction

Dynamic Node Property Prediction

➢ Predict property of a node at a future timestamp

Node affinity Prediction Task

- Predict how the preference of a user towards items change over time
- Uses Normalized Discounted Cumulative Gain (NDCG) metric



* Node classification and other tasks to be added in the future

Table 4: *Node affinity prediction* results.

Method	tgbn-trade NDCG@10 Validation	tgbn-trade NDCG@10 Test	tgbn-genre NDCG@10 Validation	tgbn-genre NDCG@10 Test	tgbn-reddit NDCG@10 Validation	tgbn-reddit NDCG@10 Test	tgbn-token NDCG@10 Validation	tgbn-token NDCG@10 Test
DyRep [42]	0.394±0.001	0.374±0.001	0.357±0.001	0.351±0.001	0.344± 0.001	0.312± 0.001	0.151± 0.006	0.141± 0.006
TGN [35]	0.395±0.002	0.374±0.001	<u>0.403</u>±0.010	<u>0.367</u>±0.058	<u>0.379</u>± 0.004	0.315± 0.020	0.189± 0.005	0.169± 0.006
persistence Fore. [36]	**0.860**	**0.855**	0.350	0.357	<u>0.380</u>	<u>0.369</u>	<u>0.403</u>	<u>0.430</u>
Moving Avg. [31]	<u>0.841</u>	<u>0.823</u>	**0.499**	**0.509**	**0.574**	**0.559**	**0.491**	**0.508**

- ➢ Persistence forecast and moving average outperforms TG methods
- ➢ A need for TG methods which focuses on node tasks

TGB

Temporal Graph Benchmark

- Website: https://tgb.complexdatalab.com/
- Documentation: https://docs.tgb.complexdatalab.com/
- Github: https://github.com/shenyangHuang/TGB
- `pip install py-tgb`
- Welcome to submit to our leaderboard.
- Contact: shenyang.huang@mail.mcgill.ca