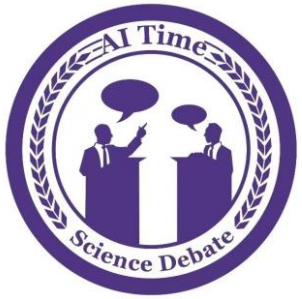
Deep Temporal Graph Clustering

Meng Liu[1] **Yue Liu**[1] **Ke Liang**[1] **Wenxuan Tu**[1]
Siwei Wang[2] **Sihang Zhou**[1] **Xinwang Liu**[1*]

[1]National University of Defense Technology, Changsha, China
[2]Intelligent Game and Decision Lab, Beijing, China
mengliuedu@163.com, xinwangliu@nudt.edu.cn

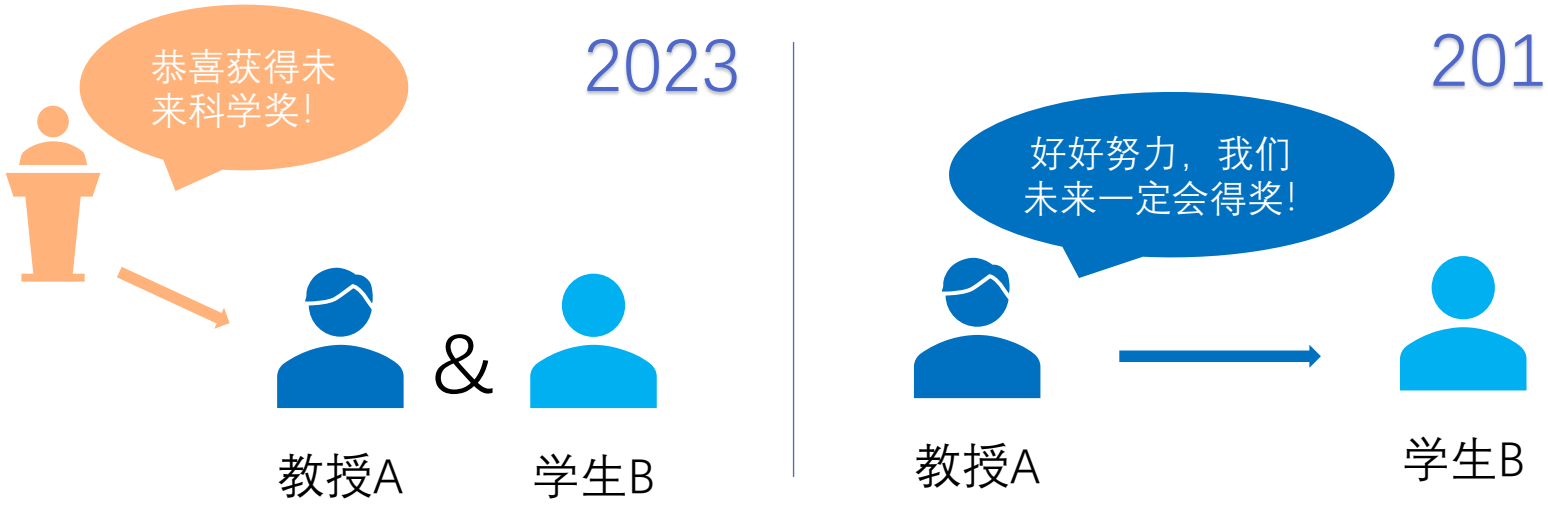


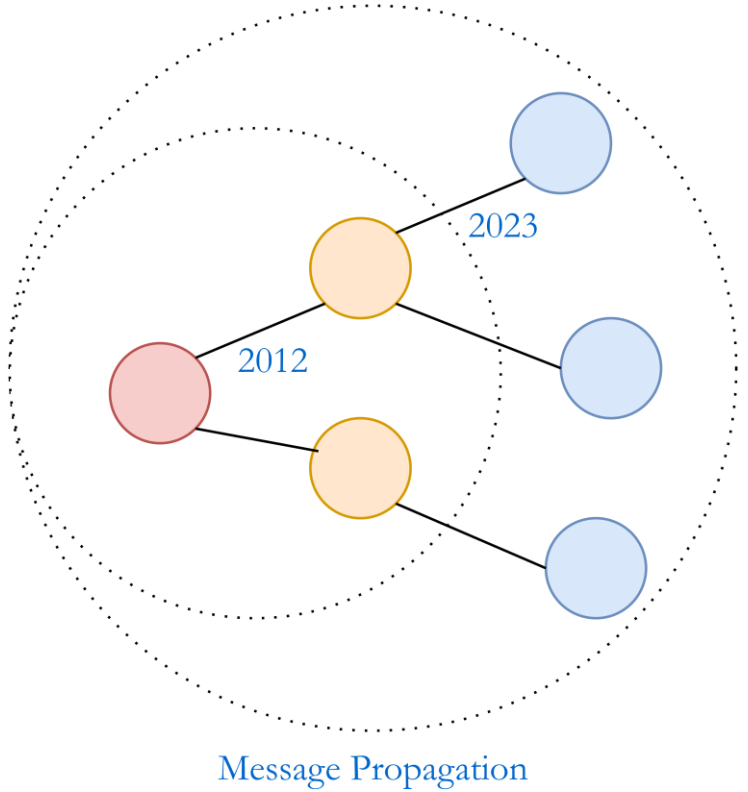
汇报人：刘猛

国防科技大学计算机学院

mengliuedu@163.com

时间的重要性

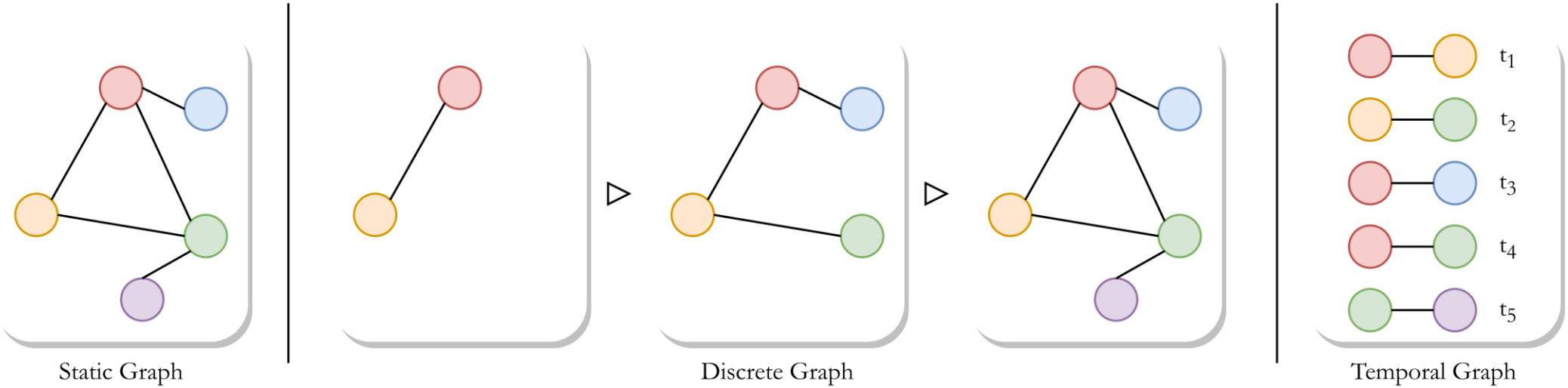




- 现实世界中，过去是无法预知未来的。
- 但图神经网络经典的消息传播机制可能会带来这个问题，即知识泄露。
- 这种情况下，时间信息就显得格外重要。

图的划分

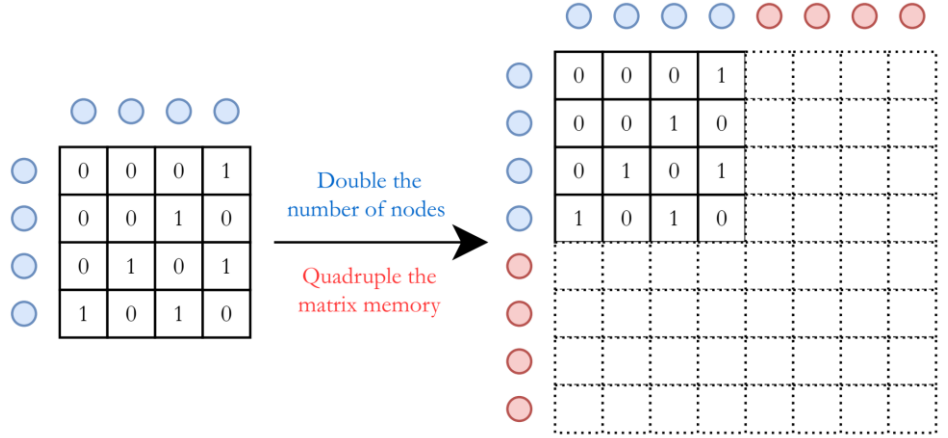
- 图数据可以根据是否含有节点交互的动态信息，分为静态图 (Static Graph) 和动态图 (Dynamic Graph)。
- 动态图又可细分为离散图 (Discrete Graph，又称Discrete-Time Dynamic Graph)和时序图 (temporal graph，又称Continuous-Time Dynamic Graph)。

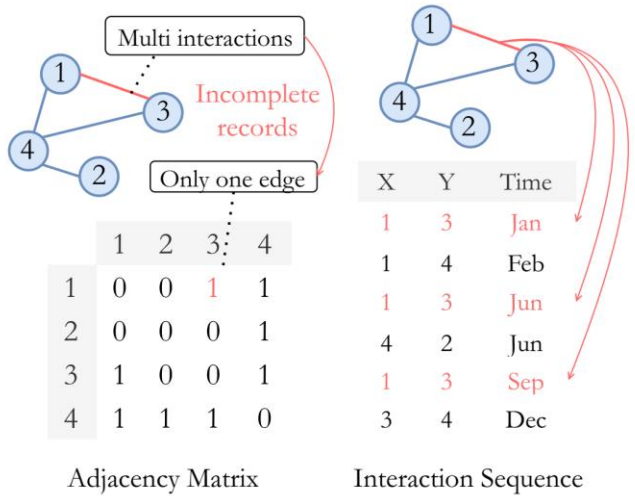


静态图、离散图和时序图

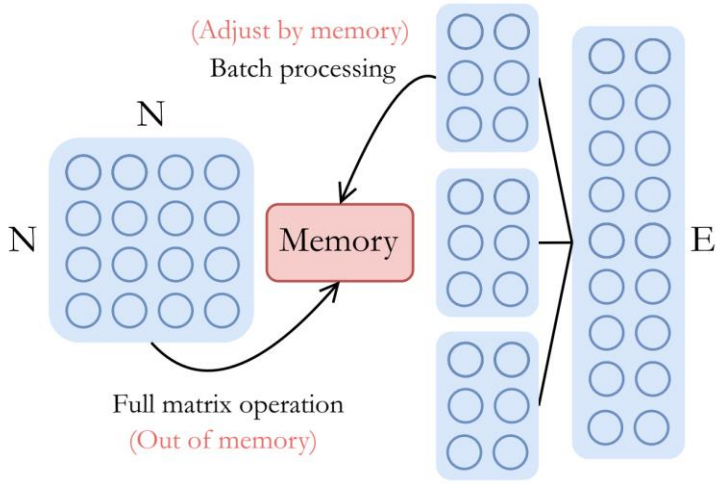
静态图与时序图

- 相比静态图基于邻接矩阵的形式，时序图采用了交互序列和批处理的模式，更加灵活，对信息的观察也更加细致。

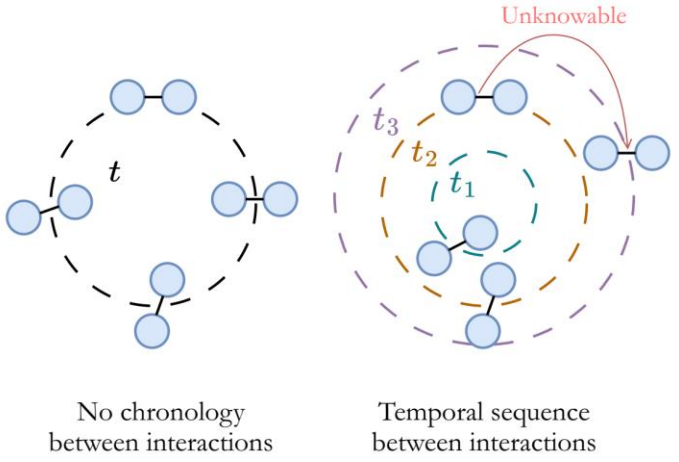




(a) Adjacency matrix and interaction sequence.



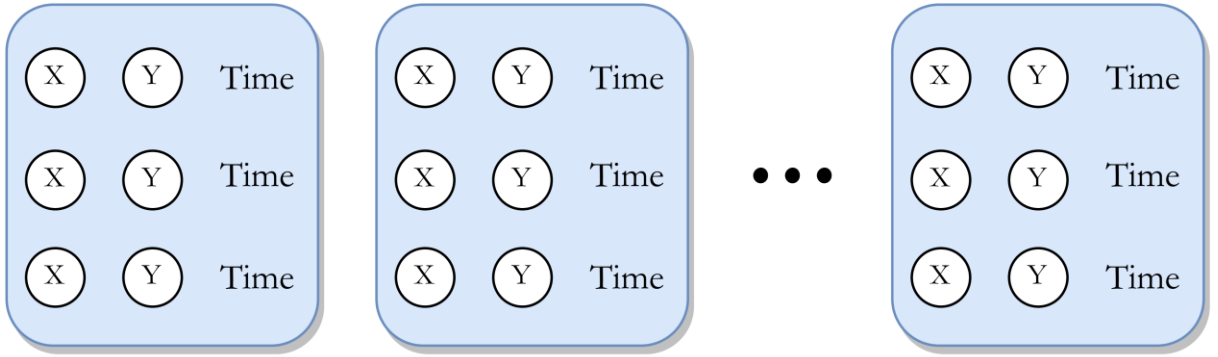
(b) Full matrix operation and batch processing.

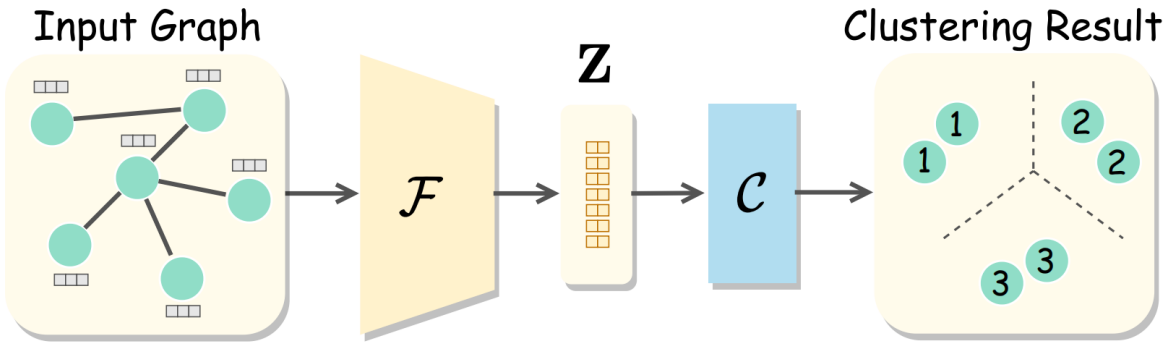


(c) Strict chronological order in temporal graphs.

困难一：聚类技术的不适用

- 时序图聚类从邻接矩阵转为交互序列的形式，带来了更丰富的数据存储结构和更方便的批处理训练模式，但随之而来的也有新的挑战。





- 脱离了邻接矩阵的数据形式，很多经典的静态图聚类技术不再适用。且脱离了邻接矩阵后，想要获取高阶结构信息变得更为困难。

TGC方法

- 我们改进了现有的经典聚类技术，将其添加到时序图学习方法上，包括节点级分布和批次级重构。
- 具体而言，我们选择HTNE作为基线方法搭载框架，但也可以迁移到其他任意时序图方法上。

节点级分布

$$q_{(x,k,t)} = \frac{(1+||\boldsymbol{z}_x^0 - \boldsymbol{z}_{c_k}^t||^2/v)^{-\frac{v+1}{2}}}{\sum_{c_j \in C}(1+||\boldsymbol{z}_x^0 - \boldsymbol{z}_{c_j}^t||^2/v)^{-\frac{v+1}{2}}}$$

$$p_{(x,k,t)} = \frac{q_{(x,k,t)}^2 / \sum_{i \in V} q_{(i,k,t)}}{\sum_{c_j \in C}(q_{(x,j,t)}^2 / \sum_{i \in V} q_{(i,j,t)})}$$

$$L_{node} = \sum_{c_k \in C} p_{(x,k,t)} \log \frac{p_{(x,k,t)}}{q'_{(x,k,t)}}$$

批次级重构

$$L_{batch} = |1 - \cos(\boldsymbol{z}_x^t, \boldsymbol{z}_y^t)| + |1 - \cos(\boldsymbol{z}_x^t, \boldsymbol{z}_h^t)| + |0 - \cos(\boldsymbol{z}_x^t, \boldsymbol{z}_n^t)|$$

损失函数

$$L_{clu} = L_{node} + L_{batch}$$

$$L = \sum^{E}(L_{tem} + L_{clu})$$

复杂度讨论

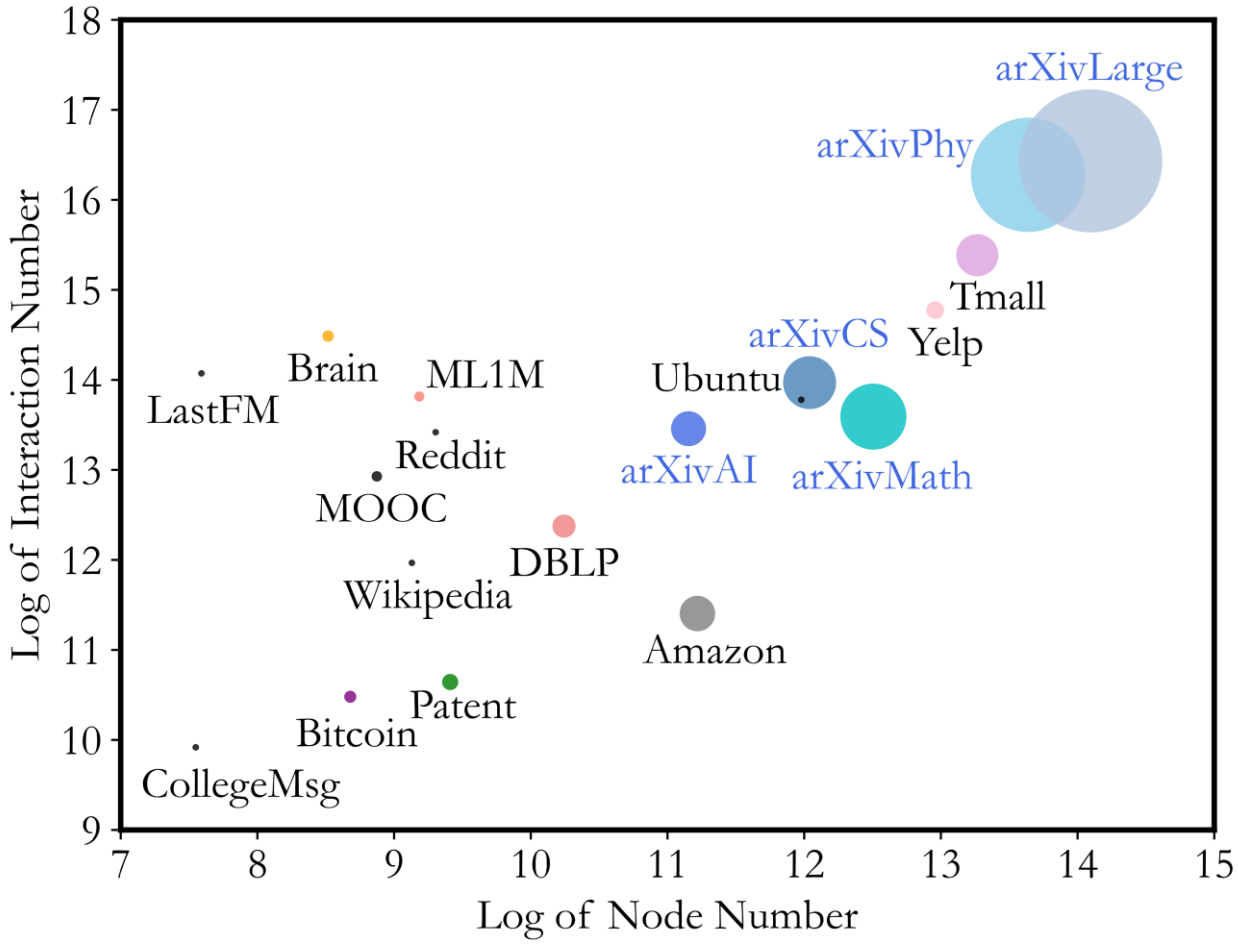
- 静态图聚类方法的核心复杂度为N^2，时序图聚类方法的核心复杂度为$|E|$。
- 大部分情况下，$N^2 > |E|$，因为N^2近似于全连接图。
- 少数情况下，$N^2 < |E|$，此时意味着边上信息的丢失或遗漏。

Table 1: Dataset statistics.

Datasets	Nodes	Interactions	Edges	Complexity	Timestamps	K	Degree	MinI	MaxI
DBLP	28,085	236,894	162,441	$N^2 \gg E$	27	10	16.87	1	955
Brain	5,000	1,955,488	1,751,910	$N^2 > E$	12	10	782	484	1,456
Patent	12,214	41,916	41,915	$N^2 \gg E$	891	6	6.86	1	789
School	327	188,508	5,802	$N^2 < E$	7,375	9	1153	7	4,647
arXivAI	69,854	699,206	699,198	$N^2 \gg E$	27	5	20.02	1	11,594
arXivCS	169,343	1,166,243	1,166,237	$N^2 \gg E$	29	40	13.77	1	13,161

困难二：数据集的不匹配

Dataset	Nodes	Interactions	Class	Labels	Timestamps
CollegeMsg	1,899	20,296	N/A	N/A	193
LastFM	1,980	1,293,103	N/A	N/A	30
Wikipedia	9,228	157,474	N/A	N/A	30
Reddit	10,985	672,447	N/A	N/A	30
Ubuntu	159,316	964,437	N/A	N/A	2,613
MOOC	7,144	411,749	N/A	N/A	-
Bitcoin	5,881	35,592	21	5,858	27,487
ML1M	9,746	1,000,209	5	3,706	25,212
Amazon	74,526	89,689	5	72,098	5,804
Yelp	424,450	2,610,143	5	15,154	153
Tmall	577,314	4,807,545	10	104,410	186
Brain	5,000	1,955,488	10	5,000	12
Patent	12,214	41,916	6	12,214	891
DBLP	28,085	236,894	10	28,085	27
arXivAI	69,854	699,206	5	69,854	27
arXivCS	169,343	1,166,243	40	169,343	29
arXivMath	270,013	799,745	31	270,013	31
arXivPhy	837,212	11,733,619	53	837,212	41
arXivLarge	1,324,064	13,701,428	172	1,324,064	41



不同数据集的规模比较

实验结果分析

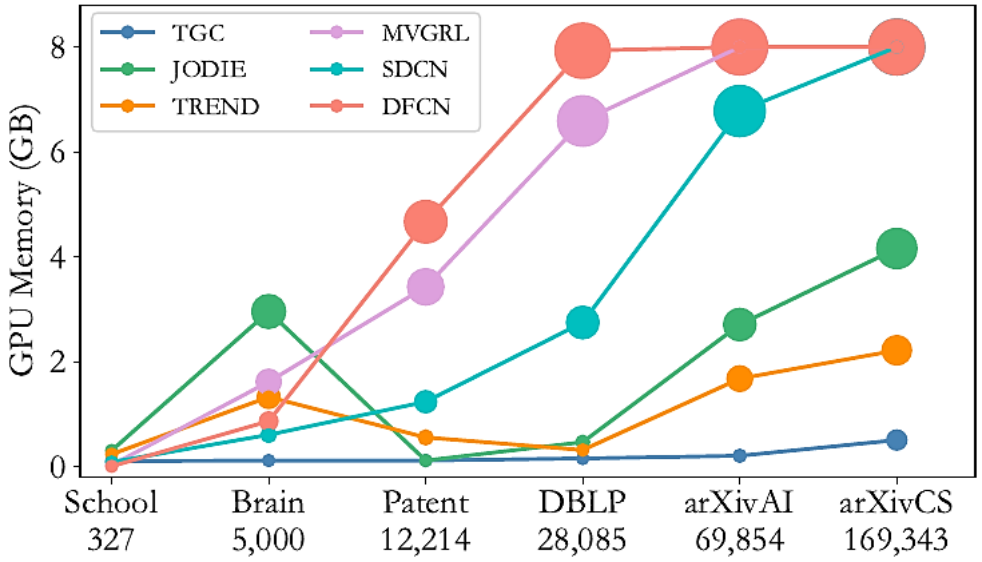
Table 2: Node clustering results in common datasets. We bold the best results and underline the second best results. If a method face the out-of-memory problem, we record as OOM.

Data	Metric	deepwalk	AE	node2vec	GAE	MVGRL	AGE	DAEGC	SDCN	SDCNQ	DFCN	HTNE	TGAT	JODIE	TGN	TREND	TGC
DBLP	ACC	28.95	42.16	46.31	39.31	28.95	OOM	OOM	46.69	40.47	41.97	45.74	36.76	20.79	19.78	46.82	**48.75**
	NMI	22.03	36.71	34.87	29.75	22.03	OOM	OOM	35.07	31.86	36.94	35.95	28.98	11.67	9.82	36.56	**37.08**
	ARI	13.73	22.54	20.40	17.17	13.73	OOM	OOM	**23.74**	19.80	21.46	22.13	17.64	11.32	5.46	22.83	22.86
	F1	24.79	37.84	43.35	35.04	24.79	OOM	OOM	40.31	35.18	35.97	43.98	34.22	13.23	10.66	44.54	**45.03**
Brain	ACC	41.28	43.48	43.92	31.22	15.76	38.48	42.52	42.62	43.42	**47.46**	43.20	41.43	19.14	17.40	39.83	44.30
	NMI	49.09	50.49	45.96	32.23	21.15	39.64	49.86	46.61	47.40	48.53	50.33	48.72	10.50	8.04	45.64	**50.68**
	ARI	28.40	29.78	26.08	14.97	9.77	28.82	27.47	27.93	27.69	28.58	29.26	23.64	5.00	4.56	22.82	**30.03**
	F1	42.54	43.26	46.61	34.11	13.56	36.47	43.24	41.42	37.27	**50.45**	43.85	41.13	11.12	13.49	33.67	44.42
Patent	ACC	38.69	30.81	40.36	39.65	31.13	43.28	46.64	37.28	32.76	39.23	45.07	38.26	30.82	38.77	38.72	**50.36**
	NMI	22.71	8.76	24.84	17.73	10.19	20.72	21.28	13.17	9.11	15.42	20.77	19.74	9.55	8.24	14.44	**25.04**
	ARI	10.32	7.43	18.95	13.61	10.26	**19.23**	16.74	10.12	7.84	12.24	10.69	13.31	7.46	6.01	13.45	18.81
	F1	31.48	26.65	34.97	30.95	18.06	35.45	32.83	31.38	28.27	30.32	28.85	26.97	20.83	21.40	28.41	**38.69**
School	ACC	90.60	30.88	91.56	85.62	32.37	84.71	34.25	48.32	33.94	49.85	99.38	80.54	65.64	31.71	94.18	**99.69**
	NMI	91.72	21.42	92.63	89.41	31.23	81.51	29.53	53.35	25.79	43.37	98.73	73.25	63.82	19.45	89.55	**99.36**
	ARI	89.66	12.04	90.25	83.09	25.00	70.24	15.38	33.81	15.82	28.31	98.70	80.04	71.94	32.12	87.50	**99.33**
	F1	92.63	31.00	91.74	82.64	24.41	84.80	31.39	45.62	33.25	47.05	99.34	79.56	68.53	29.50	94.18	**99.69**

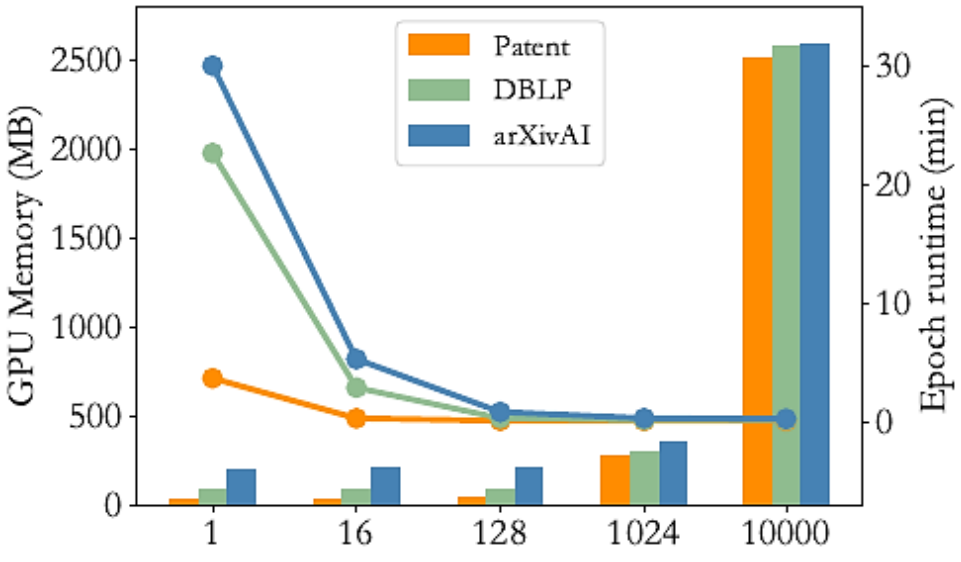
实验结果分析

Table 3: Node clustering results in large-scale datasets. We bold the best results and underline the second best results. If a method face the out-of-memory problem, we record as OOM.

Data	Metric	deepwalk	AE	node2vec	GAE	MVGRL	AGE	DAEGC	SDCN	SDCNQ	DFCN	HTNE	TGAT	JODIE	TGN	TREND	TGC
arXivAI	ACC	60.91	23.85	65.01	38.72	OOM	OOM	OOM	44.44	37.62	OOM	65.66	48.69	30.71	31.25	32.79	**73.59**
	NMI	34.34	10.20	36.18	32.54	OOM	OOM	OOM	21.63	20.73	OOM	39.24	32.12	32.16	24.74	19.82	**42.46**
	ARI	36.08	14.00	40.35	32.98	OOM	OOM	OOM	23.43	21.29	OOM	43.73	30.34	33.47	11.91	25.37	**48.98**
	F1	49.47	19.20	53.66	16.97	OOM	OOM	OOM	33.96	31.62	OOM	52.86	43.62	19.91	21.93	23.09	**57.86**
arXivCS	ACC	29.98	24.20	27.39	OOM	OOM	OOM	OOM	29.78	27.05	OOM	25.57	20.53	11.27	20.10	18.94	**39.95**
	NMI	40.86	14.03	41.18	OOM	OOM	OOM	OOM	13.27	11.57	OOM	40.83	38.64	15.50	16.21	25.58	**43.89**
	ARI	15.75	11.80	19.14	OOM	OOM	OOM	OOM	14.32	12.02	OOM	16.51	15.54	25.74	18.63	23.48	**36.06**
	F1	20.39	12.33	21.41	OOM	OOM	OOM	OOM	14.08	13.28	OOM	19.56	13.23	12.71	22.67	14.55	**25.46**



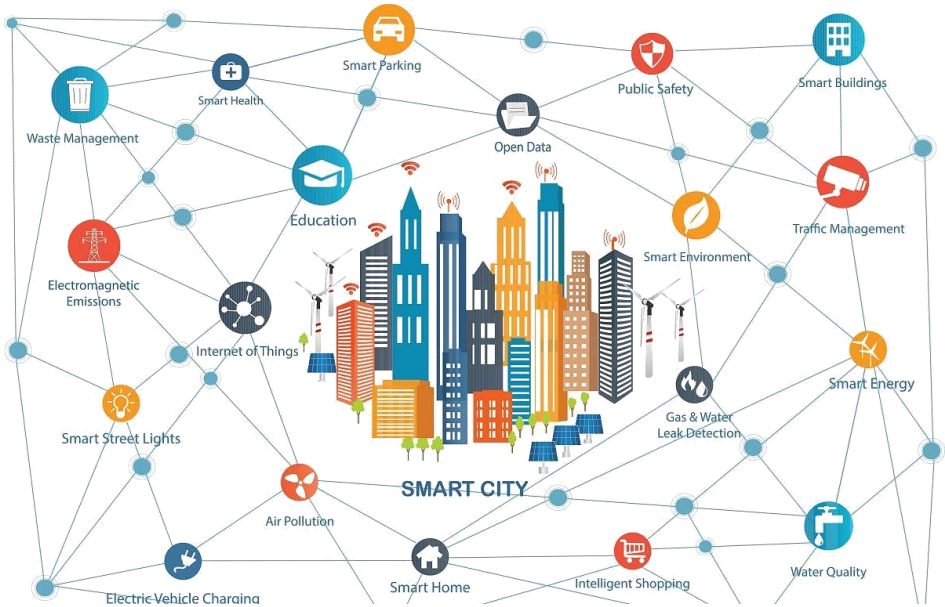
随数据规模变化的GPU内存需求

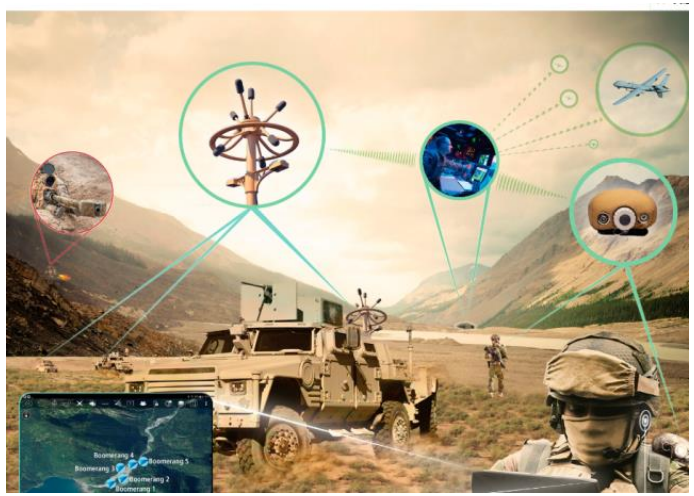


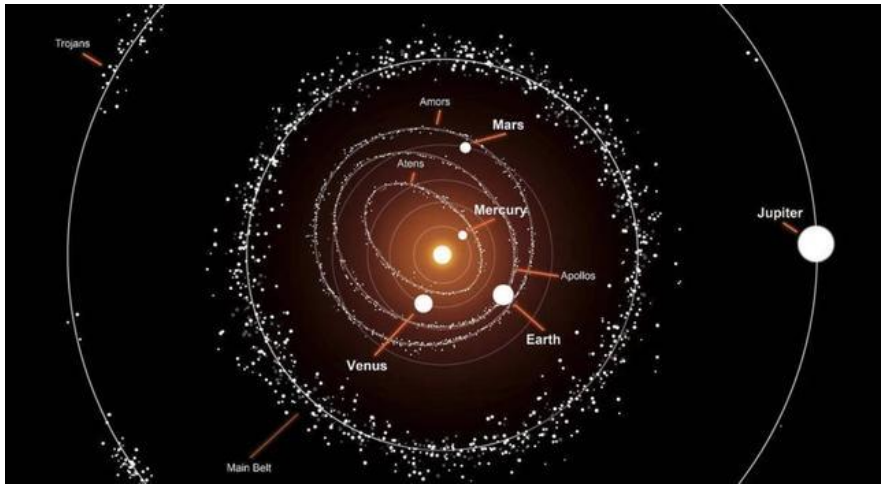
随批次变化的GPU内存和训练时间

研究总结

- 时序图聚类为深度图聚类提供了新的可能性。
- 聚类技术不适用和数据集不匹配是两大困难。
- 时序图聚类能够在时间和空间需求中找到平衡。
- 动态化的现实世界数据是时序图聚类的立足根本。







团队介绍

刘新旺，国防科技大学计算机学院教授，博士生导师。主要研究兴趣包括机器学习、数据挖掘等。

- 主持国家杰青/优青项目、“新一代人工智能”重大项目等，两次获得湖南省自然科学一等奖。
- 近五年以第一或通讯作者在CCF A类顶刊和顶会上发表论文70余篇，包括IEEE TPAMI论文10篇（含3篇独立作者），ESI高被引论文12篇。
- 谷歌学术引用15000余次，入选2022年度全球2%顶尖科学家榜单。担任IEEE TNNLS、IEEE TCYB、Information Fusion等期刊编委及ICML、NeurIPS等顶会的领域主席/资深程序委员。

刘猛，国防科技大学计算机专业博士生，导师为刘新旺教授。研究方向为时序图学习和深度聚类。

- 第一作者在ICLR、SIGIR、ACM MM、CIKM等会议期刊上发表论文7篇，谷歌学术引用200余次。
- 获CCHI 2023最佳学生论文、两次国家奖学金、校级优秀硕士论文等荣誉。担任TKDE、TOIS、TNNLS、TOMM和NeurIPS、ICML、ICLR、KDD等期刊会议审稿人。

开源项目

- Data4TGC

https://github.com/MGitHubL/Data4TGC

- Deep Temporal Graph Clustering

https://github.com/MGitHubL/Deep-Temporal-Graph-Clustering

- Awesome Knowledge Graph Reasoning

https://github.com/LIANGKE23/Awesome-Knowledge-Graph-Reasoning

Data4TGC

Data4TGC is a set of datasets for large-scale temporal graph clustering, includes DBLP, Brain, Patent, School, arXivAI, arXivCS, arXivMath, arXivPhy and arXivLarge.

This is an early version of our dataset, and we will be updating it with more information as we go along.

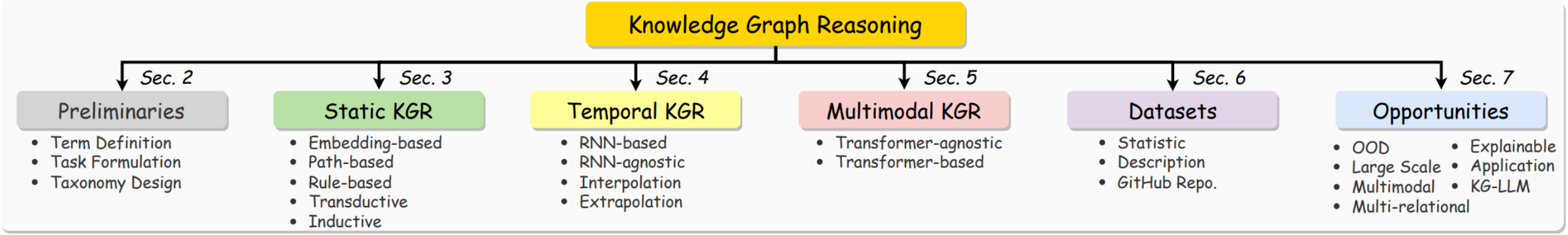
If you have any questions, please contact me: mengliuedu@163.com

Download datasets

Google Drive: https://drive.google.com/drive/folders/1-4O3V0ZcC_f8yP5ylW9CX-lE6qucbFfh?usp=sharing

Baidu Disk: https://pan.baidu.com/s/1PPgTL54Qvte7dCr0nS0vBg?pwd=1234 (Verification Code: 1234)

These downloaded datasets need to be placed under the "data" folder.



敬请各位老师批评指正!

汇报人：刘猛

mengliuedu@163.com