3rd Graph Neural Networking Workshop (GNNet '24)
December 9th, 2024

Deep Node Clustering in Attributed Temporal Graphs: Experimental Evaluation of Current Approaches

Nelson A. R. A. Passos, Emanuele Carlini, Salvatore Trani

Introduction

- **About me:**

  PhD candidate in AI @ Unipi/CNR, Italy

- **My research:**

  Intersection between Network Science and Machine Learning

- **What to expect from this presentation:**

  A discussion of the current SOTA for node-level clustering of temporal graphs

Introduction

- **About me:**

  PhD candidate in AI @ Unipi/CNR, Italy

- **My research:**

  Intersection between Network Science and Machine Learning

- **What to expect from this presentation:**

  A discussion of the current SOTA for node-level clustering of temporal graphs

  ⬇

  i.e., community detection (CD)

Introduction

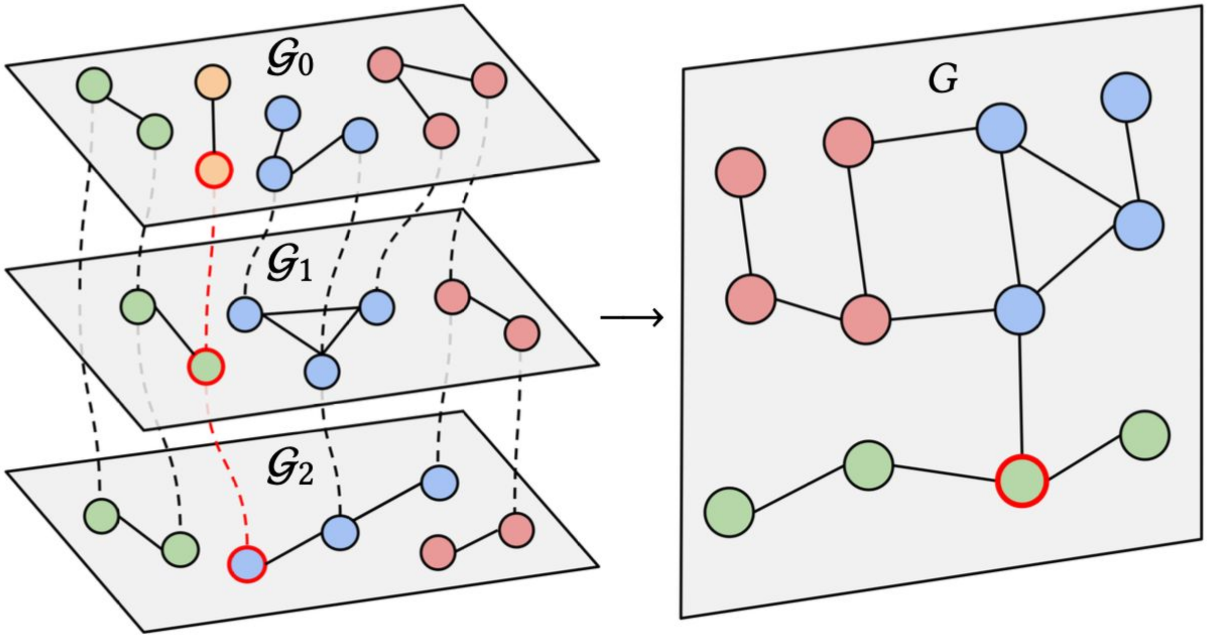
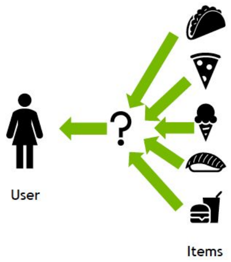
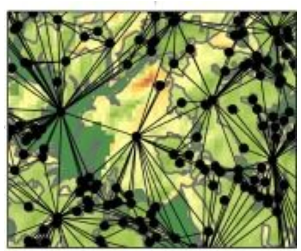


Figure: Temporal graph snapshots (left) combined in a static graph (right).

Why community detection matters?

- One of the oldest and most debated topics in Network Science
- Multidisciplinary history, with contributions from “hard” and “soft” sciences
- Multiple applications: recommendation systems, route planning and traffic control, social network analysis, wildfire detection, fraud detection + many others





Algorithmic approaches

- The first models for this task were based on the Ising model (large ongoing influence)

- Many other models introduced since then:
  - Optimization-based (modularity)
  - Statistical inference (**SBM**)
  - Matrix factorization (**NMF**)
  - Label and belief propagation (**LP/BP**)

- Non-Euclidean data: traditional ML-based approaches do not promptly work to learn on them

~~Algorithmic~~ Neural approaches

- Renewed interest in neural approaches, i.e., graph representation learning

- First models relied on "shallow" encoders:
  - **DeepWalk** (Perozzi et al., 2014)
  - **Node2Vec** (Grover et al., 2016)

- More recently: graph neural network-based models introduced for "deep" node clustering'

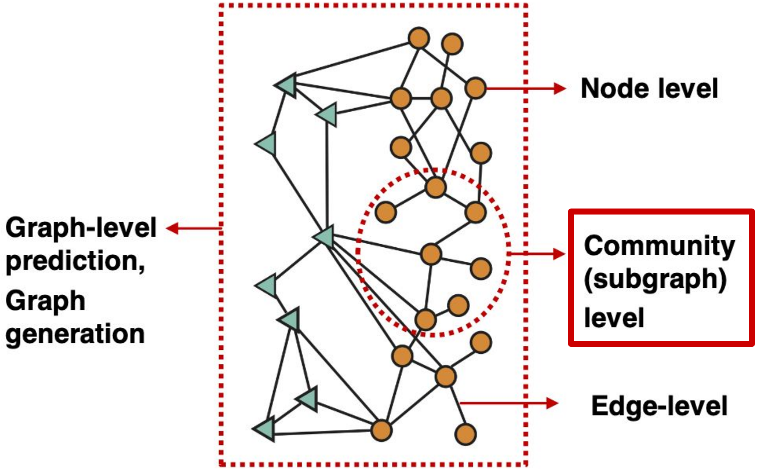


Figure: Levels of prediction of a Graph Neural Network (Leskovec et al., 2024)

What about nowadays?

- Algorithmic and neural solutions for CD are both still researched at large
- However, most models - especially GNNs - are designed for static graph learning
- In the real world, networks are rarely fixed and continuously evolve over time instead

So we asked ourselves:

How well do neural approaches for CD in temporal graphs perform when compared to more established methods?

An experimental evaluation

- We performed an evaluation of 8 models on 6 real-world datasets of various scales.
- **TGC** (Liu et al., 2024): the only GNN introduced for temporal node clustering so far

Model	Input	Topology	Features	Temporal
K-Means	X_V		✓	
Spectral Clustering	G	✓		
Leiden	G	✓		
Node2Vec	G	✓		
DynNode2Vec	$\mathcal{G}_S$	✓		✓
tNodeEmbed	$\mathcal{G}_S$	✓		✓
DAEGC	G	✓	✓	
TGC	$\mathcal{G}_E$	✓	✓	✓

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{E}\rvert$	S	d^V	y	t
arXivAI	69 854	696 819	696 819	244	128	5	27
Brain	5 000	883 207	1 007 744	1	128	10	12
DBLP	28 085	150 571	222 169	113	128	10	27
Patent	12 214	41 916	41 916	5	128	6	891
PubMed*	19 717	44 324	44 324	1	500	3	42
School	327	5 818	188 508	1	128	9	7 375

Tables 1 and 2: Models (algorithms, “shallow” encoders, graph neural networks) and datasets considered for evaluation.

Our methodology

- First we obtained node embeddings using each selected model/algorithm function.

- We then used K-Means to compare their performance and separability of embeddings.

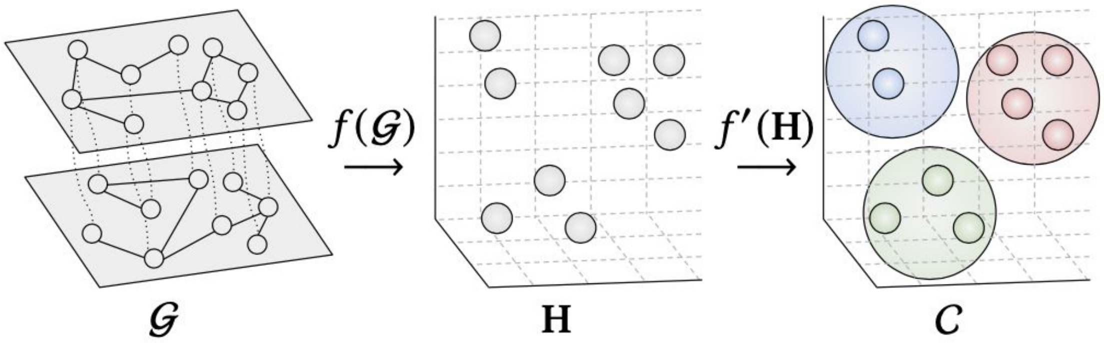


Figure: Temporal graph snapshots, node embeddings (middle), obtained clusters (right).

Transductive evaluation

Dataset	Model	ACC	AMI	ARI
arXivAI	Spectral	.389 ± .002	.016 ± .008	.012 ± .006
	Leiden	.525 ± .033	.302 ± .027	.239 ± .031
	Node2Vec	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
	DynNode2Vec	.268 ± .001	.001 ± .001	.000 ± .001
	tNodeEmbed	**.673 ± .001**	*.299 ± .001*	*.312 ± .001*
	DAEGC	OOM	OOM	OOM
	TGC	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
Brain	Spectral	**.485 ± .001**	**.498 ± .001**	**.320 ± .001**
	Leiden	.404 ± .011	.470 ± .016	.300 ± .017
	Node2Vec	*.452 ± .002*	.466 ± .001	.270 ± .001
	DynNode2Vec	.163 ± .002	.049 ± .001	.019 ± .001
	tNodeEmbed	.417 ± .002	.434 ± .001	.243 ± .002
	DAEGC	.414 ± .010	.431 ± .009	.253 ± .010
	TGC	.434 ± .001	*.497 ± .003*	*.288 ± .003*
DBLP	Spectral	.289 ± .001	.008 ± .001	.000 ± .001
	Leiden	.400 ± .013	.302 ± .003	.187 ± .005
	Node2Vec	*.466 ± .001*	*.351 ± .001*	.207 ± .001
	DynNode2Vec	.155 ± .002	.006 ± .001	.002 ± .001
	tNodeEmbed	.451 ± .001	.345 ± .001	.203 ± .001
	DAEGC	.465 ± .010	.344 ± .004	**.219 ± .004**
	TGC	**.471 ± .001**	**.355 ± .001**	*.209 ± .001*

Dataset	Model	ACC	AMI	ARI
Patent	Spectral	**.575 ± .011**	**.365 ± .023**	**.319 ± .046**
	Leiden	.431 ± .026	.203 ± .016	.140 ± .027
	Node2Vec	.400 ± .020	.270 ± .022	.171 ± .026
	DynNode2Vec	.354 ± .038	.139 ± .045	.089 ± .042
	tNodeEmbed	.424 ± .048	.248 ± .024	.177 ± .035
	DAEGC	.462 ± .029	.315 ± .052	.271 ± .052
	TGC	*.503 ± .019*	*.329 ± .027*	*.272 ± .035*
PubMed*	Spectral	.593 ± .003	.162 ± .008	.143 ± .004
	K-Means	.595 ± .001	**.312 ± .001**	.281 ± .001
	Leiden	.633 ± .018	.256 ± .020	.245 ± .033
	Node2Vec	**.687 ± .001**	.289 ± .001	**.312 ± .001**
	DynNode2Vec	.543 ± .001	.192 ± .001	.135 ± .001
	tNodeEmbed	.673 ± .001	*.297 ± .001*	*.307 ± .001*
	DAEGC	.713 ± .040	.320 ± .042	.339 ± .065
	TGC	*.677 ± .001*	.254 ± .002	.280 ± .002
School	Spectral	.967 ± .001	.940 ± .001	.933 ± .000
	Leiden	.851 ± .023	.911 ± .008	.843 ± .016
	Node2Vec	**.999 ± .002**	**.998 ± .004**	**.998 ± .004**
	DynNode2Vec	.200 ± .004	.025 ± .002	.013 ± .001
	tNodeEmbed	.200 ± .002	.025 ± .001	.013 ± .001
	DAEGC	.997 ± .006	.994 ± .009	.993 ± .010
	TGC	*.997 ± .001*	*.994 ± .001*	*.994 ± .001*

Table 3: Results comparison. Best results in **bold**, second best in *italic*, and highest mean values underlined.

Transductive evaluation

Dataset	Model	ACC	AMI	ARI
arXivAI	Spectral	.389 ± .002	.016 ± .008	.012 ± .006
	Leiden	.525 ± .033	.302 ± .027	.239 ± .031
	Node2Vec	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
	DynNode2Vec	.268 ± .001	.001 ± .001	.000 ± .001
	tNodeEmbed	**.673 ± .001**	*.299 ± .001*	*.312 ± .001*
	DAEGC	OOM	OOM	OOM
	TGC	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
Brain	Spectral	**.485 ± .001**	**.498 ± .001**	**.320 ± .001**
	Leiden	.404 ± .011	.470 ± .016	.300 ± .017
	Node2Vec	*.452 ± .002*	.466 ± .001	.270 ± .001
	DynNode2Vec	.163 ± .002	.049 ± .001	.019 ± .001
	tNodeEmbed	.417 ± .002	.434 ± .001	.243 ± .002
	DAEGC	.414 ± .010	.431 ± .009	.253 ± .010
	TGC	.434 ± .001	*.497 ± .003*	*.288 ± .003*
DBLP	Spectral	.289 ± .001	.008 ± .001	.000 ± .001
	Leiden	.400 ± .013	.302 ± .003	.187 ± .005
	Node2Vec	*.466 ± .001*	*.351 ± .001*	.207 ± .001
	DynNode2Vec	.155 ± .002	.006 ± .001	.002 ± .001
	tNodeEmbed	.451 ± .001	.345 ± .001	.203 ± .001
	DAEGC	.465 ± .010	.344 ± .004	**.219 ± .004**
	TGC	**.471 ± .001**	**.355 ± .001**	*.209 ± .001*

Dataset	Model	ACC	AMI	ARI
Patent	Spectral	**.575 ± .011**	**.365 ± .023**	**.319 ± .046**
	Leiden	.431 ± .026	.203 ± .016	.140 ± .027
	Node2Vec	.400 ± .020	.270 ± .022	.171 ± .026
	DynNode2Vec	.354 ± .038	.139 ± .045	.089 ± .042
	tNodeEmbed	.424 ± .048	.248 ± .024	.177 ± .035
	DAEGC	.462 ± .029	.315 ± .052	.271 ± .052
	TGC	*.503 ± .019*	*.329 ± .027*	*.272 ± .035*
PubMed★	Spectral	.593 ± .003	.162 ± .008	.143 ± .004
	K-Means	.595 ± .001	**.312 ± .001**	.281 ± .001
	Leiden	.633 ± .018	.256 ± .020	.245 ± .033
	Node2Vec	**.687 ± .001**	.289 ± .001	**.312 ± .001**
	DynNode2Vec	.543 ± .001	.192 ± .001	.135 ± .001
	tNodeEmbed	.673 ± .001	*.297 ± .001*	*.307 ± .001*
	DAEGC	.713 ± .040	.320 ± .042	.339 ± .065
	TGC	*.677 ± .001*	.254 ± .002	.280 ± .002
School	Spectral	.967 ± .001	.940 ± .001	.933 ± .000
	Leiden	.851 ± .023	.911 ± .008	.843 ± .016
	Node2Vec	**.999 ± .002**	**.998 ± .004**	**.998 ± .004**
	DynNode2Vec	.200 ± .004	.025 ± .002	.013 ± .001
	tNodeEmbed	.200 ± .002	.025 ± .001	.013 ± .001
	DAEGC	.997 ± .006	.994 ± .009	.993 ± .010
	TGC	*.997 ± .001*	*.994 ± .001*	*.994 ± .001*

Best results with TGC.

Table 3: Results comparison. Best results in **bold**, second best in *italic*, and highest mean values underlined.

Transductive evaluation

Dataset	Model	ACC	AMI	ARI
arXivAI	Spectral	.389 ± .002	.016 ± .008	.012 ± .006
	Leiden	.525 ± .033	.302 ± .027	.239 ± .031
	Node2Vec	*.646 ± .001*	**<u>.363 ± .001</u>**	**<u>.404 ± .001</u>**
	DynNode2Vec	.268 ± .001	.001 ± .001	.000 ± .001
	tNodeEmbed	**<u>.673 ± .001</u>**	*.299 ± .001*	*.312 ± .001*
	DAEGC	OOM	OOM	OOM
	TGC	*.646 ± .001*	**<u>.363 ± .001</u>**	**<u>.404 ± .001</u>**
Brain	Spectral	**<u>.485 ± .001</u>**	**<u>.498 ± .001</u>**	**<u>.320 ± .001</u>**
	Leiden	.404 ± .011	.470 ± .016	.300 ± .017
	Node2Vec	*.452 ± .002*	.466 ± .001	.270 ± .001
	DynNode2Vec	.163 ± .002	.049 ± .001	.019 ± .001
	tNodeEmbed	.417 ± .002	.434 ± .001	.243 ± .002
	DAEGC	.414 ± .010	.431 ± .009	.253 ± .010
	TGC	.434 ± .001	*.497 ± .003*	*.288 ± .003*
DBLP	Spectral	.289 ± .001	.008 ± .001	.000 ± .001
	Leiden	.400 ± .013	.302 ± .003	.187 ± .005
	Node2Vec	*.466 ± .001*	*.351 ± .001*	.207 ± .001
	DynNode2Vec	.155 ± .002	.006 ± .001	.002 ± .001
	tNodeEmbed	.451 ± .001	.345 ± .001	.203 ± .001
	DAEGC	.465 ± .010	.344 ± .004	**<u>.219 ± .004</u>**
	TGC	**<u>.471 ± .001</u>**	**<u>.355 ± .001</u>**	*.209 ± .001*

Dataset	Model	ACC	AMI	ARI
Patent	Spectral	**<u>.575 ± .011</u>**	**<u>.365 ± .023</u>**	**<u>.319 ± .046</u>**
	Leiden	.431 ± .026	.203 ± .016	.140 ± .027
	Node2Vec	.400 ± .020	.270 ± .022	.171 ± .026
	DynNode2Vec	.354 ± .038	.139 ± .045	.089 ± .042
	tNodeEmbed	.424 ± .048	.248 ± .024	.177 ± .035
	DAEGC	.462 ± .029	.315 ± .052	.271 ± .052
	TGC	*.503 ± .019*	*.329 ± .027*	*.272 ± .035*
PubMed★	Spectral	.593 ± .003	.162 ± .008	.143 ± .004
	K-Means	.595 ± .001	**.312 ± .001**	.281 ± .001
	Leiden	.633 ± .018	.256 ± .020	.245 ± .033
	Node2Vec	**.687 ± .001**	.289 ± .001	**.312 ± .001**
	DynNode2Vec	.543 ± .001	.192 ± .001	.135 ± .001
	tNodeEmbed	.673 ± .001	*.297 ± .001*	*.307 ± .001*
	DAEGC	<u>.713 ± .040</u>	<u>.320 ± .042</u>	<u>.339 ± .065</u>
	TGC	*.677 ± .001*	.254 ± .002	.280 ± .002
School	Spectral	.967 ± .001	.940 ± .001	.933 ± .000
	Leiden	.851 ± .023	.911 ± .008	.843 ± .016
	Node2Vec	**<u>.999 ± .002</u>**	**<u>.998 ± .004</u>**	**<u>.998 ± .004</u>**
	DynNode2Vec	.200 ± .004	.025 ± .002	.013 ± .001
	tNodeEmbed	.200 ± .002	.025 ± .001	.013 ± .001
	DAEGC	.997 ± .006	.994 ± .009	.993 ± .010
	TGC	*.997 ± .001*	*.994 ± .001*	*.994 ± .001*

TGC comparable to N2V.

Best results with TGC.

Table 3: Results comparison. Best results in **bold**, second best in *italic*, and highest mean values <u>underlined</u>.

Transductive evaluation

Dataset	Model	ACC	AMI	ARI
arXivAI	Spectral	.389 ± .002	.016 ± .008	.012 ± .006
	Leiden	.525 ± .033	.302 ± .027	.239 ± .031
	Node2Vec	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
	DynNode2Vec	.268 ± .001	.001 ± .001	.000 ± .001
	tNodeEmbed	**.673 ± .001**	*.299 ± .001*	*.312 ± .001*
	DAEGC	OOM	OOM	OOM
	TGC	*.646 ± .001*	**.363 ± .001**	**.404 ± .001**
Brain	Spectral	**.485 ± .001**	**.498 ± .001**	**.320 ± .001**
	Leiden	.404 ± .011	.470 ± .016	.300 ± .017
	Node2Vec	*.452 ± .002*	.466 ± .001	.270 ± .001
	DynNode2Vec	.163 ± .002	.049 ± .001	.019 ± .001
	tNodeEmbed	.417 ± .002	.434 ± .001	.243 ± .002
	DAEGC	.414 ± .010	.431 ± .009	.253 ± .010
	TGC	.434 ± .001	*.497 ± .003*	*.288 ± .003*
DBLP	Spectral	.289 ± .001	.008 ± .001	.000 ± .001
	Leiden	.400 ± .013	.302 ± .003	.187 ± .005
	Node2Vec	*.466 ± .001*	*.351 ± .001*	.207 ± .001
	DynNode2Vec	.155 ± .002	.006 ± .001	.002 ± .001
	tNodeEmbed	.451 ± .001	.345 ± .001	.203 ± .001
	DAEGC	.465 ± .010	.344 ± .004	**.219 ± .004**
	TGC	**.471 ± .001**	**.355 ± .001**	*.209 ± .001*

Dataset	Model	ACC	AMI	ARI
Patent	Spectral	**.575 ± .011**	**.365 ± .023**	**.319 ± .046**
	Leiden	.431 ± .026	.203 ± .016	.140 ± .027
	Node2Vec	.400 ± .020	.270 ± .022	.171 ± .026
	DynNode2Vec	.354 ± .038	.139 ± .045	.089 ± .042
	tNodeEmbed	.424 ± .048	.248 ± .024	.177 ± .035
	DAEGC	.462 ± .029	.315 ± .052	.271 ± .052
	TGC	*.503 ± .019*	*.329 ± .027*	*.272 ± .035*
PubMed*	Spectral	.593 ± .003	.162 ± .008	.143 ± .004
	K-Means	.595 ± .001	**.312 ± .001**	.281 ± .001
	Leiden	.633 ± .018	.256 ± .020	.245 ± .033
	Node2Vec	**.687 ± .001**	.289 ± .001	**.312 ± .001**
	DynNode2Vec	.543 ± .001	.192 ± .001	.135 ± .001
	tNodeEmbed	.673 ± .001	*.297 ± .001*	*.307 ± .001*
	DAEGC	.713 ± .040	.320 ± .042	.339 ± .065
	TGC	*.677 ± .001*	.254 ± .002	.280 ± .002
School	Spectral	.967 ± .001	.940 ± .001	.933 ± .000
	Leiden	.851 ± .023	.911 ± .008	.843 ± .016
	Node2Vec	**.999 ± .002**	**.998 ± .004**	**.998 ± .004**
	DynNode2Vec	.200 ± .004	.025 ± .002	.013 ± .001
	tNodeEmbed	.200 ± .002	.025 ± .001	.013 ± .001
	DAEGC	.997 ± .006	.994 ± .009	.993 ± .010
	TGC	*.997 ± .001*	*.994 ± .001*	*.994 ± .001*

Methods **outperforming** TGC.

TGC **comparable** to N2V.

Best results with TGC.

Table 3: Results comparison. Best results in **bold**, second best in *italic*, and highest mean values underlined.

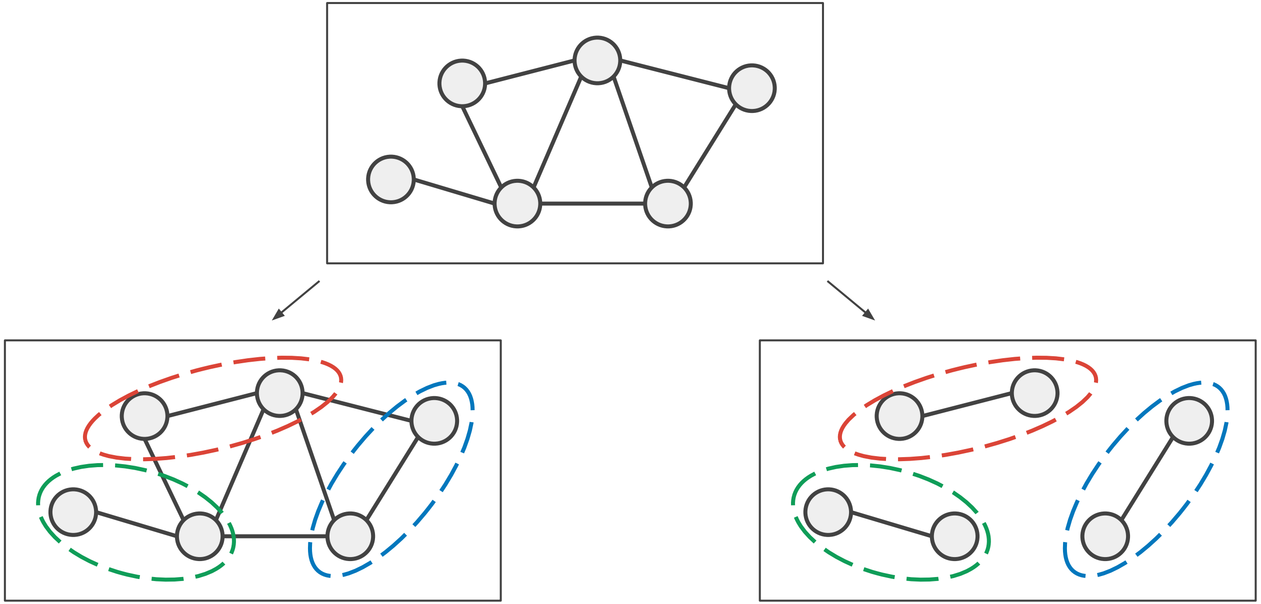
Transductive vs. inductive

- Most GNNs for CD are evaluated in a transductive learning setting only

- This is mostly due to a lack of datasets for the task
  - Temporal graph data
  - With node-level features
  - With community ground truths

- According to some authors, this restricts evaluation to an “overfitting competition”

Transductive vs. inductive



Transductive split.

Inductive split.

– – Training set.

– – Validation set.

– – Test set.

Figure: Graph learning settings and node-level splits.

Transductive vs. inductive

- We therefore were interested in expanding our evaluation to an inductive setting, but this was only possible for a single dataset we constructed from PubMed data:

Dataset	Model	ACC	AMI	ARI
PubMed	K-Means	**.682 ± .001**	*.235 ± .002*	**.272 ± .001**
	DAEGC	*.690 ± .010*	**.257 ± .012**	*.285 ± .020*
	TGC	.678 ± .002	.221 ± .005	.254 ± .002

Table 4: Results comparison for inductive learning setting.
Best results in **bold**, second best in *italic*, and highest mean values underlined.

- Since the node features for the other datasets we used were obtained by the original authors by pretraining with Node2Vec, we could not prevent information leakage for them

Results breakdown

- GNNs yielded the best possible results only on **one out of six** datasets here evaluated
- There is still a **large room for improvements** on neural methods for community detection
- Although useful for many tasks, they are still not the de-facto state of the art for this task
- We need **more research**: more datasets, models, and interest in neural community detection
    - Especially in attributed temporal graphs, due to the detectability threshold of communities

Concluding remarks

- **Research opportunities:** network scientists and ML researchers both agree that the detectability threshold of communities can be improved by exploiting temporality/features

- A GNN-based model may be one of the best candidates for this goal!

- Such a model would likely benefit many real-world applications, ranging from e-commerce to environmental studies, from traffic prediction to research in social network dynamics.

- We aim to continue with our research in this direction, especially for inductive learning

Summary: Our contributions

- Experimental evaluation of **algorithmic and neural node clustering methods** (8 models, 6 real-world datasets, transductive + inductive learning when possible)

- **PubMedTemporal**: newly released temporal edge data and node-level temporal split (available from Zenodo/GitHub and soon from within PyTorch Geometric)

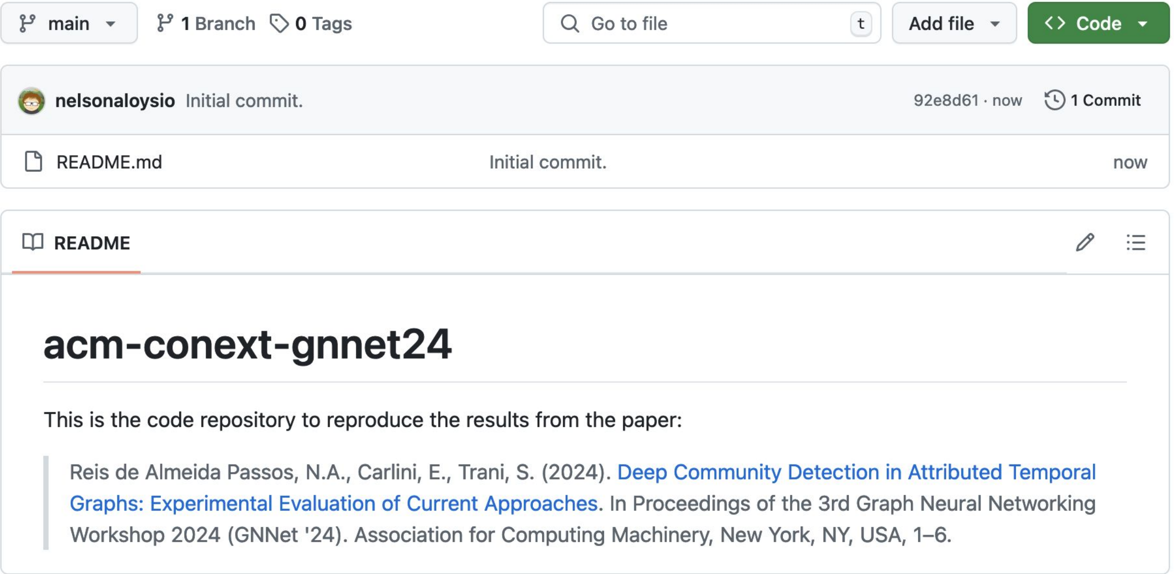
- **Code reproducibility**: available from GitHub to foster further research in the area

Acknowledgements

A huge thank you to the following research groups for their valuable support:

- High Performance Computing laboratory of ISTI-CNR (Pisa, Italy)
- Department of Network and Data Science of the CEU (Vienna, Austria)
- Inverse Complexity Lab of the IT:U (Linz, Austria)

Thank you!



Code repository:

github.com/nelsonaloysio/gnnet24

nelson.reis@phd.unipi.it nelsonaloysio