TADC-SBM: a Time-Varying, Attributed, Degree-Corrected Stochastic Block Model

Passos, N.A.R.A.; Carlini, E.; Trani, S.

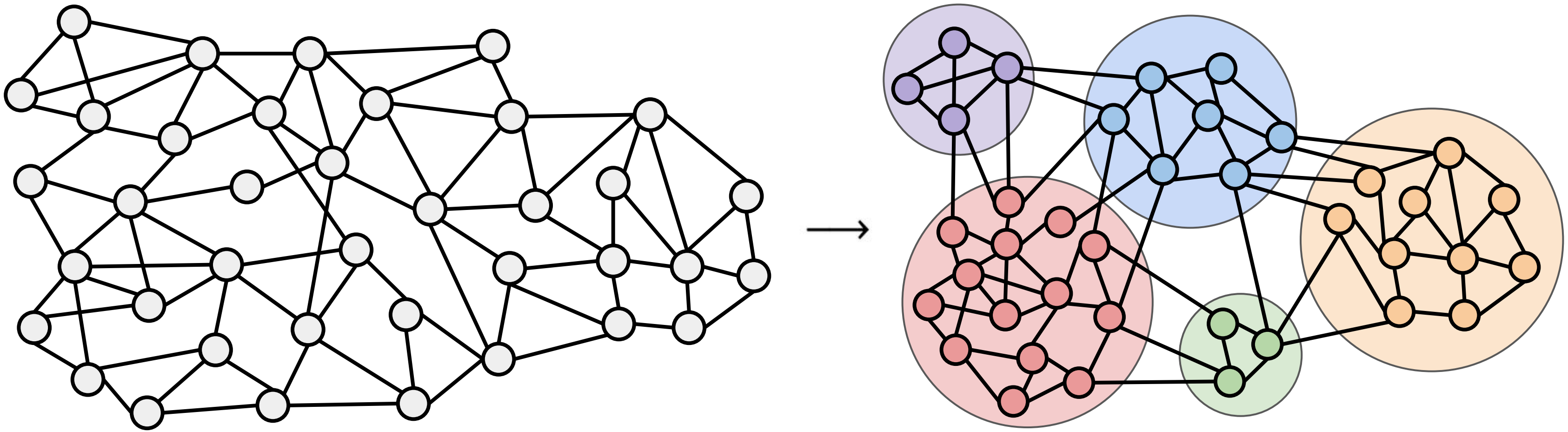
July 4th, 2025

Outline

- Community detection
- Why “neural” community detection?
- Datasets (real-world vs. synthetic)
- Model benchmarking
- Experimental evaluation
- Results/Conclusions

Community detection

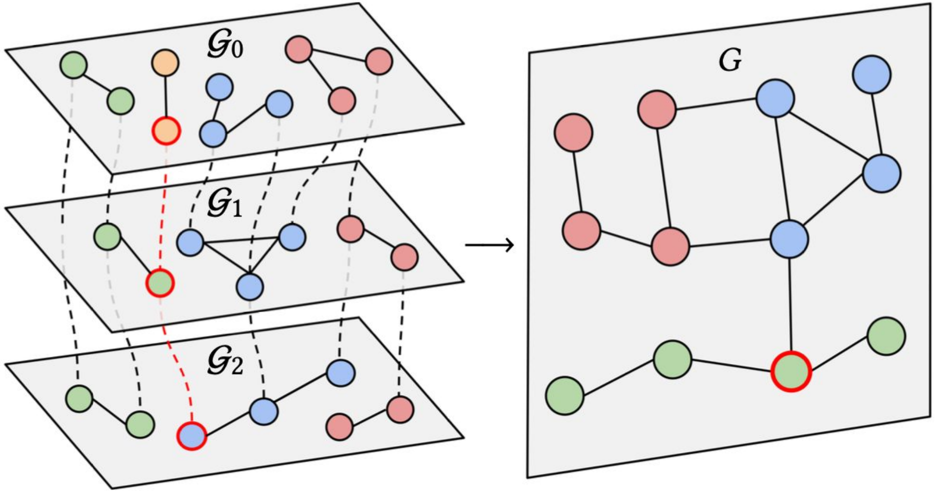
- A paradigmatic task in **network science** is partitioning a network graph into node subsets.



A graph with $|V|$=38 nodes and $|E|$=81 edges (62 within-community edges and 19 out-edges).

Community detection

- A paradigmatic task in **network science** is partitioning a network graph into node subsets.
- Temporally evolving graphs naturally add another layer of complexity to this task.



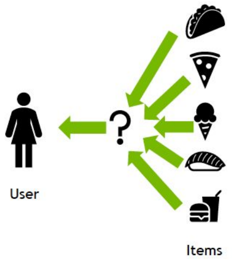
Temporal graph snapshots (left) combined in a static graph (right).

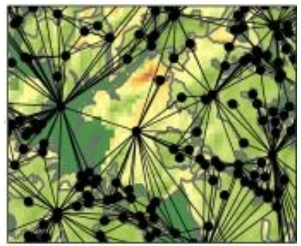
Community detection

- A paradigmatic task in **network science** is partitioning a network graph into node subsets.

- Temporally evolving graphs naturally add another layer of complexity to this task.

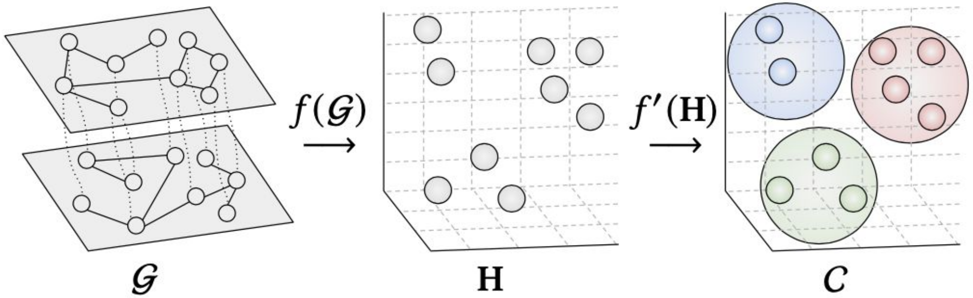
- **Multiple domain applications**:
    - recommendation systems;
    - route planning and traffic control;
    - fraud and anomaly detection;
    - social network analysis;
    - biochemistry/functional analysis;
    - wildfire prevention;
    - and many others.





Why "neural" community detection?

- Graph representation learning models allow exploiting a graph's (*i*) topology, (*ii*) temporal dynamics, and (*iii*) attribute features to obtain node, edge, or graph-level embeddings.



A function f fits (*learns*) a graph $\mathcal{G}$ and maps nodes to embeddings $\mathbf{H}$, which are then used to obtain a set $\mathcal{C}$ of communities (*clusters*) [1].

- This joint exploration potentially improves on the detectability thresholds [2] of the graph's communities, while the obtained functions (*models*) may be used to predict unseen data.
- Real-world graphs for model evaluation are the norm in AI research → **but a *flawed* one**!

---

[1] Passos et al., ACM CoNEXT/GNNet Workshop, 2024.
[2] Nadakuditi & Newman, Phys. Rev. Letters, 2012.

Why not real-world graphs?

- Real-world temporal graph data is scarce, of limited scope, and ground truths are dubious [3].

  <u>**Scarce**</u> → few datasets are available, so the model that best overfits them wins; most datasets are too limited or costly to fully explore the relative performances of GNNs [4].

  <u>**Limited scope**</u> → most available datasets are either citation or communication networks, thus narrowing the assessment of how useful those models are in other domains.

  <u>**Dubious ground truths**</u> → node labels (classes/communities) come from handcrafted, domain-specific categories that may hold little relation w. graph topology/attributes.

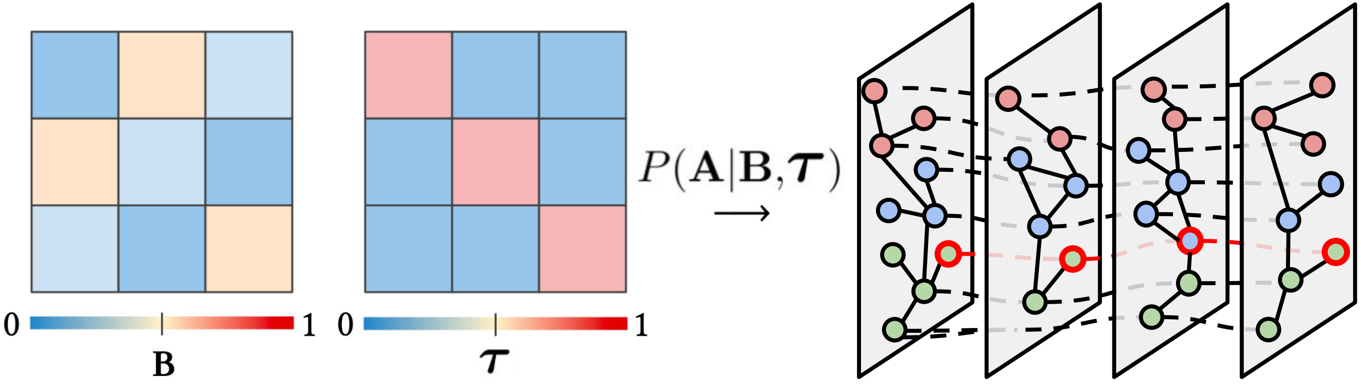
- In sum: "***there are no planted communities in*** (temporal) ***real-world networks***" [3].

---

[3] Peel et al., Science Advances, 2017.
[4] Palowitch et al., 28th ACM SIGKDD, 2022.

How to evaluate those models?

- To overcome it, we introduce the **TADC-SBM** generator, a Time-varying, Attributed, Degree--Corrected Stochastic Block Model [4] based on [5, 6] for benchmarks in *controlled* scenarios.



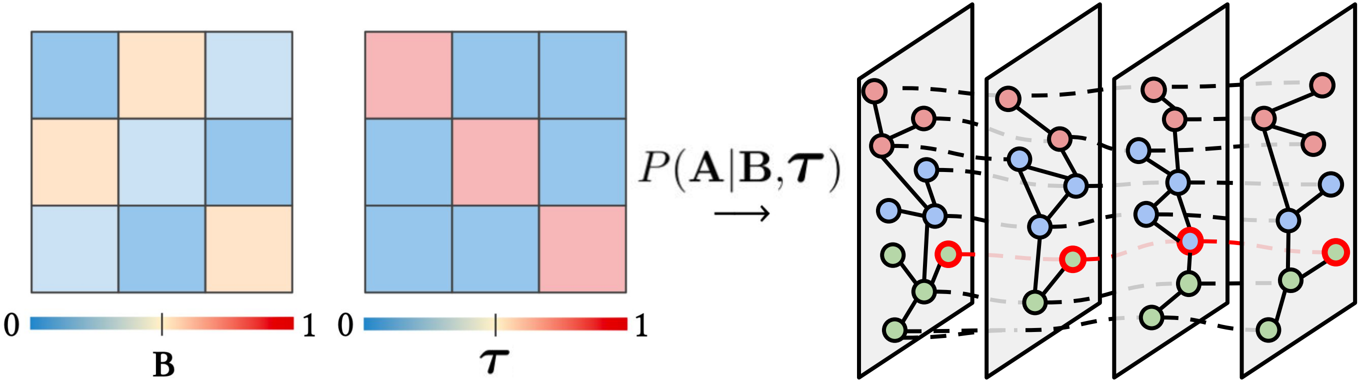
In addition to a block matrix **B**, we employ a transition matrix $\boldsymbol{\mathcal{T}}$ to control the probability of nodes transitioning communities over time.

- This principled approach allows to compare different temporal community detection models.

[3] Peel et al., Science Advances, 2017.
[5] Ghasemian et al., Phys. Rev. X, 2016.
[6] Tsitsulin et al., ACM Web Conference/GLB Workshop, 2021.

Our experimental setup

- We focused on the "special" case where $\boldsymbol{\tau} := \eta\,\mathbf{I} + (1-\eta)\,\dfrac{\mathbf{J}-\mathbf{I}}{k-1}$ for our experiments.



In addition to a block matrix $\mathbf{B}$, we employ a transition matrix $\boldsymbol{\tau}$ to control the probability of nodes transitioning communities over time.

- Nodes have a uniform-at-random chance of $1-\eta$ of switching communities per snapshot.

- Additional parameters β = [0,1] controls edge sampling and γ = {0, 1} fixes transitions.

Details and metrics

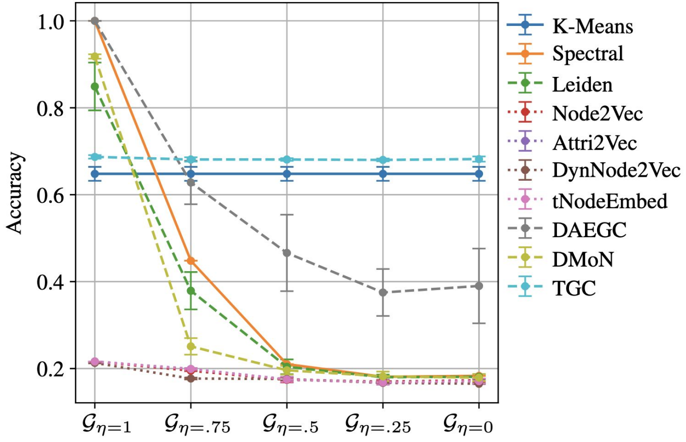
- Using our model, we first generated graphs with k=8 **clusters and** t=8 **snapshots**.
- We varied only the transition probability $\eta \in \{0, 0.25, 0.5, 0.75, 1\}$ for each graph.
- Node **attributes** (s = 32-dimensional features) are generated **once** per node/community.
- Edge distribution follows a power law (α = 2) with expected total of $|E| = d \times |V|$ edges.
- Average node degree approximates $\langle d \rangle = (d + (k-1)\, d^{\star})/k$, where d = 20.
- Additional parameters β = [0,1] controls edge sampling and γ = {0, 1} fixes transitions.

Dataset	Model	Accuracy	AMI	ARI
$\mathcal{G}_{\eta=1}$	K-Means	.648 ± .016	.400 ± .015	.375 ± .018
	Spectral	**1.000 ± .000**	**1.000 ± .000**	**1.000 ± .000**
	Leiden	.849 ± .055	*.945 ± .022*	*.848 ± .048*
	Node2Vec	.216 ± .000	.066 ± .000	.041 ± .000
	Attri2Vec	.216 ± .000	.066 ± .000	.041 ± .000
	DynNode2Vec	.213 ± .001	.060 ± .002	.037 ± .001
	tNodeEmbed	.216 ± .000	.066 ± .000	.041 ± .000
	DAEGC	**1.000 ± .000**	**1.000 ± .000**	**1.000 ± .000**
	DMoN	*.918 ± .005*	.813 ± .011	.815 ± .011
	TGC	.687 ± .004	.438 ± .005	.421 ± .005
$\mathcal{G}_{\eta=.75}$	K-Means	*.648 ± .016*	*.400 ± .015*	*.375 ± .018*
	Spectral	.448 ± .000	.152 ± .000	.135 ± .000
	Leiden	.379 ± .043	.132 ± .016	.115 ± .017
	Node2Vec	.195 ± .001	.023 ± .001	.014 ± .000
	Attri2Vec	.199 ± .002	.026 ± .001	.017 ± .000
	DynNode2Vec	.177 ± .002	.012 ± .002	.006 ± .001
	tNodeEmbed	.199 ± .002	.026 ± .001	.017 ± .000
	DAEGC	.628 ± .050	.356 ± .040	.337 ± .055
	DMoN	.251 ± .019	.051 ± .007	.062 ± .007
	TGC	**.681 ± .005**	**.434 ± .006**	**.415 ± .007**
$\mathcal{G}_{\eta=.5}$	K-Means	*.648 ± .016*	*.400 ± .015*	*.375 ± .018*
	Spectral	.210 ± .000	.025 ± .000	.018 ± .000
	Leiden	.204 ± .017	.019 ± .008	.012 ± .005
	Node2Vec	.174 ± .006	.007 ± .004	.004 ± .002
	Attri2Vec	.175 ± .006	.005 ± .004	.003 ± .002
	DynNode2Vec	.176 ± .003	.005 ± .001	.003 ± .000
	tNodeEmbed	.175 ± .006	.005 ± .004	.003 ± .002
	DAEGC	.466 ± .088	.218 ± .050	.180 ± .058
	DMoN	.196 ± .010	.014 ± .004	.026 ± .003
	TGC	**.681 ± .003**	**.432 ± .005**	**.415 ± .005**

Results for graphs with higher stability rates.

Benchmarking results

- Model performance overall severely degraded as $\eta \to 0$.
- **Exception:** the TGC [7] model displayed good resilience.
- Traditional (*algorithmic*) approaches performed better or as good as SOTA neural models in most scenarios.
- The same in a previous study [1] with real-world graphs.



Limitations and future work

- Extending the model to support mixed memberships.
- Generating dynamic (node and edge-level) features.
- Further evaluate graph embedding and statistical models.

← Preprint and code

nelsonaloysio.github.io

[1] Passos et al., ACM CoNEXT/GNNet Workshop, 2024. [7] Liu et al., ICLR, 2024.