NetSciX2026: International School and Conference on Network Science
February 17^{th}-20^{th}, 2026 - Auckland, New Zealand

Efficient "Neural" Community Detection in Attributed Graphs with a Temporal Modularity Loss Function

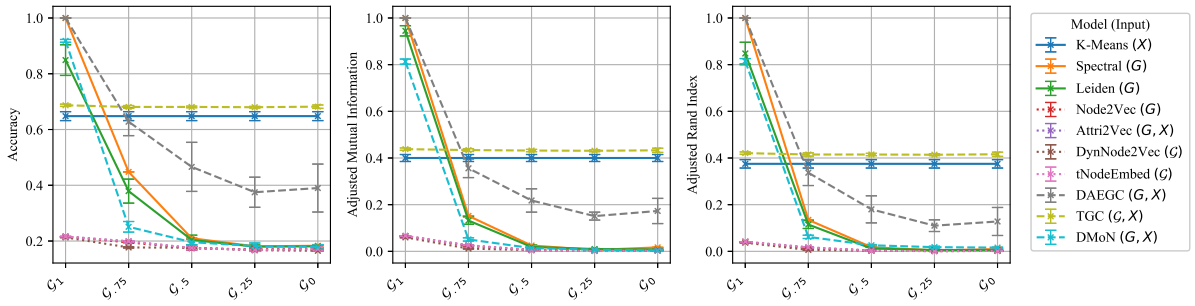
Keywords: Community Detection, Temporal Networks, Graph Neural Networks, Modularity

Extended Abstract

The compendium of community detection techniques has been significantly enriched in recent years by the inclusion of neural network-based architectures for graphs. Although these models have demonstrated state-of-the-art performance for node, link, and graph-level prediction tasks, in applications such as recommendation systems, anomaly detection, and protein design, their utility for community detection, particularly in dynamic graphs, warrants further investigation. We build upon previous work on "neural" community detection in static graphs and extend an efficient end-to-end solution [1] to the temporal setting, in which the loss function optimizes a temporal (longitudinal) version of the classic modularity metric on the graph spectral domain.

L-DMoN learns a (differentiable) function $f : \mathscr{G} \to \mathbf{C}_{(n,k)} \in [0,1]$ by spectral modularity maximization on a dynamic graph $\mathscr{G} = (V,E,\mathbf{X})$, where V is the set of n nodes, E is the set of m temporal edges, and $\mathbf{X}_{(n,d)}$ is a matrix of d-dimensional node attributes (feature vectors). Neural network-based approaches for node classification appeal to real-world applications due to their scalability and expressiveness, as they are able to learn complex patterns from graphs, avoiding the need for manual feature engineering. However, node and edge attributes may hold little relation to the community structure of the network, particularly in temporal settings where nodes may transition between communities. In contrast with other neural approaches, the use of unsupervised loss functions that directly optimize a graph-theoretic "quality" metric help to mitigate this issue by providing a more objective (descriptive) measure of community structure.

Within this framework, node memberships are obtained by relaxing the NP-hard problem of discrete modularity maximization into a continuous one, in which a message-passing function receives both the graph structure and node features as input, and outputs low-dimensional representations followed by a normalized exponential function. The community assignment matrix is computed as $\mathbf{C} = \mathrm{softmax}(\psi(\tilde{\mathbf{A}}, \mathbf{X}^{(\mathbf{l})}))$, where ψ is a (possibly multilayer) graph convolutional encoder, $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix, and $\mathbf{X}^{(l)}$ are the node features at layer l. Embeddings are computed by $\mathbf{X}^{(l+1)} = \phi(\tilde{\mathbf{A}}\mathbf{X}^{(l)}\mathbf{W} + \mathbf{X}\mathbf{W}_{\text{skip}})$, where ϕ is a non-linear activation function (SeLU), $\mathbf{W}$ and $\mathbf{W}_{\text{skip}}$ are learnable weight matrices (the latter with trainable skip connections in place of self-loops), and $\mathbf{X}^{(0)} = \mathbf{X}$, i.e., the input node features. Lastly, dropout is applied between layers to avoid local optima on gradient descent.



Graph datasets $\mathcal{G}_\eta$ with varying signal-to-noise ratio η in [0, 1]. Legend: algorithms (solid line), graph encoders (dotted) and GNNs (dashed). Input: $\mathcal{G}$, G are temporal or static graphs and X are node features.

Figure 1: **Preliminary benchmarks.** Five synthetic datasets $\mathscr{G}_\rho$ were generated from a degree-corrected stochastic block model with $|V| = 1024$ nodes, $|E| = 10^4$ edges, $|T| = 8$ time steps, $d = 32$ node features, and varying signal-to-noise ratios (SNR) introduced by nodes transitioning communities. The static model (DMoN) initially displays good performance, but degrades as SNR decreases ($\rho \to 0$).

The loss function $\mathscr{L}$ is based on longitudinal modularity [2], a recent time-aware extension of the classical metric that abstracts away the need for temporal segmentation (slicing the graph into snapshots) and incentivizes communities with similar temporal signatures. For efficiency, a spectral relaxation of the metric is employed, which is maximized by leveraging sparse-matrix operations and rank-one normalization, with complexity scaling with $O(d^2n+m)$. The model is encouraged to find communities with stable (smooth) temporal dynamics, while the trivial solution of assigning all nodes to a single community is avoided by adding an inexpensive collapse regularization term. In a high level, $\mathscr{L}$ may be expressed in terms of three components: the spectral longitudinal modularity Q, temporal smoothness S, and collapse regularization R,

$$\mathscr{L} = Q + S + R = -\frac{1}{2m}\mathrm{Tr}(\mathbf{C}^{\mathrm{T}}\mathbf{B}\mathbf{C}) + \frac{\omega}{2m}\mathrm{Tr}(\mathbf{C}^{\mathrm{T}}\mathbf{S}\mathbf{C}) + \frac{\vartheta\sqrt{k}}{n}\left|\left|\sum_i C_i^{\mathrm{T}}\right|\right|_F - 1,$$

where $\mathbf{B}$ is the modularity matrix, $\mathbf{S}$ is the smoothness matrix, k is the number of communities, $\omega \in \mathbb{R}^+$ and $\vartheta \in \mathbb{R}^+$ are smoothing and regularizer parameters with a default value of 1, and $||\cdot||_F$ is the Frobenius norm of the cluster membership counts, normalized in the range $[0, \sqrt{k}]$ and equal to zero when cluster sizes are perfectly balanced. The modularity term Q is negative as we minimize $\mathscr{L}$ via gradient descent, while S is equivalent to that defined in [2] in case of hard assignments, i.e., $S = \frac{\omega}{2m}\sum_u \eta_u$, where η_u is the "community switch count" of node u over time, which is here adapted to the spectral domain with mixed memberships (soft clusters). We note that recent architectural extensions [3, 4] propose more sophisticated regularization strategies that may be adapted to the temporal setting, as well as contrastive learning techniques that aim to enhance cluster separability in the feature space, which we have yet to explore.

Temporal dynamics are considered by defining $\mathbf{B} = \mathbf{A}^\star - \mathbf{P}$, where $\mathbf{A}^\star$ is the (weighted, directed) adjacency matrix and $\mathbf{P}$ is a null model matrix that accounts for temporal node activity. Following the mean-membership longitudinal link expectation model [2], which expects node interactions to be proportional to the lifetimes of nodes within each community, we hence have

$$\mathbf{B}_{(n\times n)} = \mathbf{A}^\star - \frac{1}{|\tau|}\frac{\langle \mathbf{d}_{\mathrm{out}}\,\mathbf{p}\rangle^{\mathrm{T}}\langle \mathbf{d}_{\mathrm{in}}\,\mathbf{p}\rangle}{\mathbf{C}^{\mathrm{T}}\mathbf{A}^\star\mathbf{C}}, \quad \mathbf{S}_{(n\times n)} = \mathbf{I}_n\left[\sum_{t_i\in\tau_u} 1\,\mathrm{if}\,(t_{i+1}-t_i > \delta)\,\middle|\,\forall u \in V\right]$$

where τ and τ_u is the set of graph and node u time steps, $\mathbf{d}_{\mathrm{out}}$ and $\mathbf{d}_{\mathrm{in}}$ are the (weighted) out- and in-degree vectors, $\mathbf{p}$ is the vector of node presences $p_u = |\tau_u|^{\frac{1}{2}}/|\tau| \in [0,1]$, i.e., the geometric mean of their lifetimes, $\mathbf{I}$ is the identity matrix, and δ is a threshold (1 for discrete time steps).

As far as we aware, this is the first neural approach proposed to optimize a temporal version of modularity in an end-to-end fashion. The presented solution is designed to be efficient and scalable to large graphs, while the spectral version of longitudinal modularity is novel and may independently be of interest to the community. An additional challenge of note we aim to tackle is employing a time-aware graph pooling layer to obtain dynamic community assignments, so to allow nodes to transition between communities over time. We will further experiment with and share our implementation online to foster research, ensure fairness and reproducibility, and properly assess the model's limitations w.r.t. varying detectability regimes on presentation.

References

[1] Anton Tsitsulin et al. "Graph Clustering with Graph Neural Networks". In: *Journal of Machine Learning Research* 24.127 (2023), pp. 1–21.

[2] Victor Brabant et al. "Longitudinal modularity, a modularity for link streams". In: *EPJ Data Science* 14.1 (2025).

[3] Yasmin Salehi and Dennis Giannacopoulos. "Deep Modularity Networks with Diversity-Preserving Regularization". In: *CoRR* abs/2501.13451 (2025).

[4] Yunfei Liu et al. "Revisiting Modularity Maximization for Graph Clustering: A Contrastive Learning Perspective". In: *Proc. of the 30th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. KDD '24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 1968–1979.