Learning and Clustering on Temporal Graphs: Principles, Primitives, and Pooling

Nelson Aloysio Reis de Almeida Passos[1,2][0000−0003−1869−2976], Emanuele Carlini[2][0000−0003−3643−5404], and Salvatore Trani[2][0000−0001−6541−9409]

[1] University of Pisa, Dept. of Computer Science, 56127 Pisa PI, Italy
[2] National Research Council, 56124 Pisa PI, Italy

nelson.reis@phd.unipi.it
{emanuele.carlini,salvatore.trani}@isti.cnr.it

This work[3] focuses on the problem of learning on temporal graphs, with a particular emphasis on the principles and primitives surrounding the task of *clustering* [17]: obtaining coarse-grained representations by aggregating information from nodes, edges, and temporal dynamics — a task related to that of *pooling* [2] in machine learning on graphs, or *community detection* [14] in network science.

We depart from a question intersecting both literatures: whether learning on graphs and community detection reinforce one another, compete, or prove largely orthogonal in practice. Graph neural networks (GNNs) have reached state of the art for diverse downstream tasks, yet their advantage over established descriptive and inferential clustering algorithms [14] remains far less settled [5], particularly under joint demands of computational efficiency and recovery accuracy [13].

Our central claim is that temporal clustering becomes relevant to graph learning when treated simultaneously as a statistical *principle*, a scalable computational *primitive*, and a *pooling* mechanism. An initial study of ours found that augmenting neural models with temporal structure did not consistently improve the recovery of ground-truth communities over algorithmic baselines [8]; a finding later corroborated on synthetic graphs sampled, under known ground truth, from a time-varying, attributed, degree-corrected stochastic block model we designed for this purpose [10]. For non-attributed temporal community detection, principled algorithmic methods remain the appropriate tool, and the decisive obstacle is scalability rather than accuracy; the case for neural models is strongest precisely in the attributed regime that descriptive spectral and modularity [7] methods do not exploit in full, and even there the gains are clearest where structural, attribute, and temporal signals align, rather than as a universal advantage [14].

Principles. Community detection rests on the premise that nodes organize into groups whose connectivity — and, where present, attributes and temporal behavior — set them apart from the rest of the network [14]. The prevailing notion is assortative: groups more densely connected within than without, the definition underpinning descriptive quality functions such as modularity and relaxations solved on graph spectra [7]. A sharper, theory-grounded criterion is the *detectability* of communities: in the sparse stochastic block model (SBM)

[3]Accepted to ECML PKDD 2026 (Nectar Track), based on prior work [8,10,11,12].

there exists a threshold — the Kesten–Stigum bound $c\lambda^2 = 1$, with λ the community signal and c the average degree — below which no efficient algorithm recovers planted structure better than chance in the large-graph limit [1]. Ordinary Laplacian spectra fall short of this bound, their informative eigenvectors localizing on high-degree nodes; operators built on the non-backtracking (Hashimoto) matrix instead remain informative down to the threshold itself [15]. The same theory extends to the temporal domain, where a second parameter $\eta \in [0, 1]$ — the rate at which nodes retain their community label across snapshots — couples spatial and temporal signal, so that detectability comes to depend jointly on λ and η [1].

Crucially, spectral theory also underlies graph learning. A node signal $\mathbf{x}$ is projected to the spectral domain as $\hat{\mathbf{x}} = \mathbf{U}^\top \mathbf{x}$, with $\mathbf{U}$ the Laplacian eigenbasis, so that a learned filter acts as $\mathbf{y} = \mathbf{U} g_\theta(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{x}$; graph convolutions are localized polynomial approximations of this construction, aggregating over K-hop neighborhoods without explicit eigendecomposition [3]. So community detection and graph learning draw on a common foundation: the operators that reveal mesoscale structure are those on which message passing is defined. It is this principled footing that motivates spectral and modularity-based methods as the substrate for end-to-end differentiable clustering [17] — a choice that, as we show, has practical consequences for a GNN's performance, not merely theoretical ones.

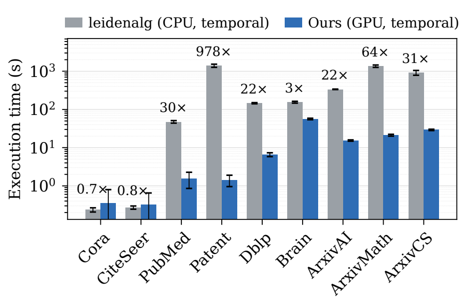
Primitives. Algorithmic methods are traditionally CPU-bound and ill-suited to large temporal graphs: temporal coupling of nodes through time enlarges the adjacency matrix to order $nT \times nT$, so cost grows prohibitively with both the number of nodes and the number of time steps [6]. Our recent contribution makes them tractable at scale — GPU-accelerated multislice modularity optimization [6,16] and spectral formulations [7,15] over sparse supra-adjacency representations, built on the NVIDIA RAPIDS (cuGraph, cuML) ecosystem [11]. However, the spectral case is non-trivial on GPUs: accelerated sparse eigensolvers are currently restricted to symmetric (real) or Hermitian (complex) operators, whereas the non-backtracking matrix attaining the detectability threshold on temporal graphs is, by construction, asymmetric — aptly, as information flows forward in time.

We escape the asymmetry by reformulating the eigenproblem: solving the symmetric Bethe-Hessian $\mathbf{H}(r) = (r^2 - 1)\mathbf{I} - r\mathbf{A} + \mathbf{D}$, whose negative eigenvalues recover the informative, out-of-bulk directions of the non-backtracking operator [15] — preserving asymptotic optimality for undirected graphs while keeping matrix operations fully on GPU. In static SBM graphs, $r \approx \sqrt{c}$, while the temporal case requires coupling λ and η [1] — hence a different regularization, as likely does the attributed regime. Against a CPU reference under an equal-work budget, our multislice modularity backend attains speedups of up to roughly three orders of magnitude, depending on graph density and snapshot count (Figure 1). For the largest graphs, the effect is a change in tractability: temporal graphs that exceed practical time limits on CPU become routine on GPU. Exposed as a zero-code-change backend of the NetworkX-Temporal library [9], the implementation requires only a single environment variable to move a pipeline from CPU to GPU.

Pooling. This primitive is of interest beyond community detection itself. Learning on very large graphs is increasingly limited by scale, and a standard response is coarse-graining: reducing the graph to a smaller representation on which a model can operate, then refining or unpooling the result [2]. This reduction is rarely inconsequential — most graph pooling is heuristic or learned without guarantees — and a principled community detector offers an alternative: a coarse-graining operator grounded in detectability theory, where the temporal supra-graph assigns labels jointly across snapshots, dispensing with post-hoc matching between independently detected partitions [6]. Spectral modularity assigns communities by the trace $\mathrm{Tr}(\mathbf{C}^\top \mathbf{M}\,\mathbf{C})$ of a partition matrix $\mathbf{C}$ against the modularity matrix $\mathbf{M}$ [7]; the same bilinear form $\mathbf{C}^\top(\cdot)\,\mathbf{C}$ is exactly the pooling operation that differentiable clustering objectives may optimize end-to-end [17].

A principled partition $\mathbf{C}$ is thus already a pooling assignment, and a scalable, detectability-optimal way to obtain it is a pooling primitive for graphs too large to train on directly. That these objectives optimize the same block form does not make them interchangeable: differentiable relaxations are sensitive to how the partition is regularized, and a poorly chosen constraint can drive the pooled assignments below graph-theoretic baselines [17] — precisely where such a partition would provide a stable starting point for learning representations.

Our work identifies two complementary directions for future research. One is *neural*: improving differentiable temporal pooling by grounding it in detectability theory — principled rather than heuristic — while retaining its block form. The other is *algorithmic*: attaining the detectability threshold at scale — through GPU eigensolvers for asymmetric matrices, dispensing with the symmetric reformulation entirely, or through a temporal regularization recovering the same informative directions. Underlying both is a question we raise for the machine learning and network science communities: when does community structure suffice as a basis for learning on coarse-grained representations, and what more do tasks depending on temporally ordered dependencies rather than co-membership alone require? Characterizing when community-based pooling preserves these dependencies in the attributed temporal context is, in our view, the open problem this research must strive to answer, and the one toward which it is now directed.



Dataset	$\lvert\mathcal{V}\rvert$	$\Sigma(\mathcal{V})$	$\lvert\mathcal{E}\rvert$	$\Sigma(\mathcal{E})$	$\lvert T\rvert$
Cora †	2 708	2 708	5 278	5 278	1
CiteSeer †	3 279	3 279	4 552	4 552	1
PubMed	19 717	37 003	44 324	44 324	42
Patent	12 214	41 529	41 916	41 916	891
Dblp	28 085	101 797	153 822	222 165	27
Brain	5 000	60 000	878 207	947 744	12
ArxivAI	69 854	142 316	696 819	696 819	27
ArxivMath	270 013	614 578	783 165	783 165	31
ArxivCS	169 343	374 433	1 157 799	1 157 799	29

Fig. 1. Runtime comparison [11]. Left: CPU/GPU runtimes over five runs, mean ± standard deviation (log scale). Right: Datasets [4,8]: $\lvert\mathcal{V}\rvert, \lvert\mathcal{E}\rvert$ are unique nodes/edges, $\Sigma(\mathcal{V}), \Sigma(\mathcal{E})$ are sums over snapshots, and $\lvert T\rvert$ is the snapshot count; † marks static baselines. Matched two-iteration Leiden budget (`leidenalg` default) on an Intel Xeon Gold 6330 (CPU) or NVIDIA A100 80GB (GPU), including host-to-device transfer.

References

1. Ghasemian, A., Zhang, P., Clauset, A., Moore, C., Peel, L.: Detectability thresholds and optimal algorithms for community structure in dynamic networks. Physical Review X **6**(3) (7 2016)
2. Grattarola, D., Zambon, D., Bianchi, F.M., Alippi, C.: Understanding pooling in graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **35**(2), 2708–2718 (2024)
3. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th Int. Conference on Learning Representations, ICLR (2017)
4. Liu, M., Liang, K., Liu, Y., Wang, S., Zhou, S., Liu, X.: arXiv4TGC: Large-scale datasets for temporal graph clustering (2023)
5. Longa, A., Lachi, V., Santin, G., Bianchini, M., Lepri, B., Liò, P., Scarselli, F., Passerini, A.: Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. Transactions on Machine Learning Research (2023)
6. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. Science **328**(5980), 876–878 (5 2010)
7. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006)
8. Passos, N.A.R.A., Carlini, E., Trani, S.: Deep community detection in attributed temporal graphs: Experimental evaluation of current approaches. In: Proceedings of the 3rd Graph Neural Networking Workshop 2024. pp. 1–6. GNNet ’24, Association for Computing Machinery, New York, NY, USA (2024)
9. Passos, N.A.R.A., Carlini, E., Trani, S.: NetworkX-Temporal: Building, manipulating, and analyzing dynamic graph structures. SoftwareX **31**, 102277 (2025)
10. Passos, N.A.R.A., Carlini, E., Trani, S.: TADC-SBM: a time-varying, attributed, degree-corrected stochastic block model. In: 2025 IEEE Symposium on Computers and Communications (ISCC). pp. 1–6 (2025)
11. Passos, N.A.R.A., Carlini, E., Trani, S.: Accelerating dynamic graph clustering on GPU architectures with cuGraph (2026), accepted to: the 6th Workshop on Flexible Resource and Application Management on the Edge (FRAME ’26), co-located with Euro-Par ’26, 24–28 August, 2026, Pisa, Italy
12. Passos, N.A.R.A., Carlini, E., Trani, S.: Toward efficient community detection on time-varying attributed graphs (2026), accepted to: Doctoral Symposium of the 32nd International European Conference on Parallel and Distributed Computing (Euro-Par ’26), 24–28 August, 2026, Pisa, Italy
13. Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. Science Advances **3**(5) (5 2017)
14. Peixoto, T.P.: Descriptive vs. Inferential Community Detection in Networks: Pitfalls, Myths and Half-Truths. Elements in the Structure and Dynamics of Complex Networks, Cambridge University Press (2023)
15. Saade, A., Krzakala, F., Zdeborová, L.: Spectral clustering of graphs with the Bethe Hessian. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in NIPS. vol. 27. Curran Associates, Inc. (2014)
16. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports **9**(1), 5233 (3 2019)
17. Tsitsulin, A., Palowitch, J., Perozzi, B., Müller, E.: Graph clustering with graph neural networks. Journal of Machine Learning Research **24**(127), 1–21 (2023)