BIAS-DRIVEN GRAPH TRANSFORMERS WITH MEMORY

ZHANG Xinjie

Advisors: Miao Wang, Ali Hirsa, Noah Dawang

Columbia University

Motivation & Problem

Dynamic graphs (finance, biology, social systems) evolve, yet downstream analysis benefits from *temporal stability*:

- Identify subgraphs that persist across windows while capturing new interactions.
- Maintain consistent cluster identities over time.
- Preserve continuity in low-dimensional visualizations.

Method: learn *bias-driven* attention augmented with *node/edge memory* to obtain robust embeddings; construct stable subgraphs; perform rolling clustering with label alignment and stability monitoring.

Graph Construction & Biases

- **Base graph:** standardize features and build a correlation k-NN graph (retain top-n_{nbr} positive correlations as weighted edges).
- **History-aware edges:** when Jaccard(neighborhood$_{t-1}$, neighborhood$_t$) exceeds a threshold, re-introduce prior edges with weight 0.7 (tagged as memory edges).
- **Node positional encodings** p_i: fixed-width hop histogram $(1 \dots H)$ plus k Laplacian eigenvectors (zero-padded); standardized to a constant dimension.
- **Node similarity** S: cosine similarity on standardized p_i, rescaled to $[0, 1]$.
- **Edge structural kernel** $K(i, j) = \exp(-\alpha\, d(i, j))$ from shortest-path distance.
- **Learned fusion** of S and K under non-negativity reparameterization:

$$B_{ij} = \begin{cases} S^{w_{\text{node}}} \cdot K^{w_{\text{edge}}} & \text{product} \\ \dfrac{w_{\text{node}} S + w_{\text{edge}} K}{w_{\text{node}} + w_{\text{edge}}} & \text{weighted sum} \end{cases}$$

- **Bias-informed attention & edge attributes:** dense B induces a scalar edge attribute for graph transformer layers; robust numerical guards are applied and self-loops receive unit attribute as fallback.

Subgraph Partitioning

- **Thresholded selection** with learned $(\tau_{\text{att}}, \gamma_{\text{ker}}, \tau_{\text{hm}}, \kappa)$: node sets by *norm*, *attention*, or their union/intersection; edges retained if $K(i, j) > \gamma_{\text{ker}}$.
- **Node/edge bias modes:** degree, betweenness, closeness, eigenvector centrality; kernel-thresholded edges.
- **Combined graph cut:** $W = \lambda_{\text{att}} \cdot \text{Att}_{\text{bias}} + w_{\text{node}} S + w_{\text{edge}} K$, followed by spectral clustering on W.

Example Visualization

Bias-Driven Graph Transformer with Memory

Backbone: three-layer graph transformer with edge attributes from B_{ij}; an output head produces node embeddings h (with nonlinearities and dropout).

Memory (dimension $= 32$):

- **Node memory**

$$m_i^t \leftarrow \alpha\, m_i^{t-1} + (1 - \alpha) \cdot \text{Linear}\Big(\text{mean}\{h_j^t : j \in N(i)\}\Big).$$

- **Edge memory**

$$m_{ij}^t \leftarrow \beta\, m_{ij}^{t-1} + (1 - \beta) \cdot \text{Linear}\big([h_i^t \| h_j^t]\big).$$

- **Attention input:** when memory is used, $z_i = [h_i \,\|\, \lambda m_i]$ informs bias-aware attention and subgraph selection.

Cross-time identifiers & matching: persistent node/edge labels (ϕ, ψ); similarity via Weisfeiler–Lehman kernel, isomorphism checks (VF2), and graph edit cues. A frequency-based *core* is formed and expanded at time t; newly formed edges are marked as dynamic.

Optimization & Temporal Training

Loss components:

- Structural margin ranking (positive vs. negative edges) and neighbor smoothness.
- Attention stability: $1 - \dfrac{\sum \min(A_t, A_{t-1})}{\sum \max(A_t, A_{t-1})}$ on bias-aware attention A_t.
- Memory regularization: deviation from node-wise running means.
- Coverage control: selection-rate regularization via learned thresholds $(\tau_{\text{hm}}, \gamma_{\text{ker}}, \kappa)$.

Training proceeds per-window or in short backpropagation-through-time segments.

Spectral Loss

From bias-aware attention $A \in \mathbb{R}^{N \times N}$ (sanitized), optionally symmetrize $A \leftarrow \frac{1}{2}(A + A^\top)$ and clamp $A \geq 0$. Define

$$L_{\text{sym}} = I - D^{-1/2} A D^{-1/2}, \qquad D = \text{diag}(A\mathbf{1}).$$

Let $Z \in \mathbb{R}^{N \times k}$ collect k non-trivial eigenvectors of L_{sym} (ignoring eigenvalues $\leq \varepsilon$; $k = \min(\text{clusters}, N - \text{start})$). The objective is

$$\mathcal{L}_{\text{spec}} = \text{tr}\big(Z^\top L_{\text{sym}} Z\big),$$

used alone or combined with the structural objectives.

Clustering & Label Stability

- **Methods: R^2 Kmeans**, standard K-means (warm-start from prior centroids), spectral (nearest-neighbors embedding with K-means), Louvain, and attention-based grouping.
- **Label alignment:** Hungarian matching where applicable (e.g., Louvain/attention partitions); K-means uses warm starts.
- **Quality:** silhouette score is reported when well-defined.

Evaluation: Stability Metrics

High-dimensional ($t \to t{+}1$):

- Jaccard overlap of nodes/edges for subgraph unions.
- Density and average clustering.
- Distance between normalized-Laplacian spectra.
- Top-k neighborhood preservation (by attention/edge weights).
- Node/edge memory variance; per-window silhouette (if valid).

Low-dimensional (DR on $z = [h \| \lambda m]$):

- KNN Jaccard between consecutive windows.
- High→Low neighborhood preservation (KNN in z vs. DR).
- Temporal variance of low-dimensional embeddings.

R^2 Kmeans

A temporally stateful variant of K-means for sequential windows.

- **Cross-time initialization:** seeds are drawn from active centroids of the previous window, prioritized by cluster size.
- **Windowed centroid history:** each cluster maintains a short chain of recent centroids, together with counts and last-known locations.
- **Sparse-case adaptation ($n_{\text{unique}} < k$):** deactivate or remove distant centroids; reactivate plausible historical ones; introduce centroids at distinct data points when necessary; adjust k accordingly.
- **Label assignment:** collapsed centroid chains provide stable labels at transform time.
- **Distance/preprocessing:** supports multiple metrics and normalization to improve robustness across windows.

Quantitative Comparison

	Stability (↑)				Entropy (↓)			
Method	**Orig.**	**Ours**	**Δ**	**Rel.**	**Orig.**	**Ours**	**Δ**	**Rel.**
Vanilla Spectral Clustering	0.651	0.651	0.000	0.0%	1.493	1.493	0.000	0.0%
Memory KMeans	0.678	1.000	0.322	47.5%	1.443	0.000	-1.443	N/A
Memory R^{2}KMeans	0.360	0.987	0.627	173.9%	1.447	1.505	-0.058	-4.0%
Memory Louvain	0.881	0.889	0.008	0.9%	0.997	0.985	0.012	1.2%
Bias-based	0.808	0.818	0.010	1.3%	0.634	0.601	0.033	5.2%

Acknowledgements

This work builds upon the R^2 Spectral Clustering framework developed by Miao Wang, Ali Hirsa, Satyan Malhotra, and Noah Dawang.