Graph Variate Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

Modelling dynamically evolving spatio-temporal signals is a prominent challenge in the Graph Neural Network (GNN) literature. Notably, GNNs assume an existing underlying graph structure. While this underlying structure may not always exist or is derived independently from the signal, a temporally evolving *functional* network can always be constructed from multi-channel data. Graph Variate Signal Analysis (GVSA) defines a unified framework consisting of a network tensor of instantaneous connectivity profiles against a stable support usually constructed from the signal itself. Building on Graph-Variate Signal Analysis (GVSA) and tools from graph signal processing, we introduce **Graph-Variate Neural Networks (GVNNs)**: layers that convolve spatio-temporal signals with a signal-dependent connectivity tensor combining a stable long-term support with instantaneous, data-driven interactions. This design captures dynamic statistical interdependencies at each time step without ad-hoc sliding windows and admits an efficient implementation with linear complexity in sequence length. Across forecasting benchmarks, GVNNs consistently outperform strong graph-based baselines and are competitive with widely used sequence models such as LSTMs and Transformers. On EEG motor-imagery classification, GVNNs achieve strong accuracy highlighting their potential for brain–computer interface applications.

1 Introduction

The modeling of graph signals has been a pervasive topic in recent years in Graph Signal Processing (GSP) and Graph Neural Networks (GNN) (Xu et al., 2019; Kenlay et al., 2020) with a lack of a general consensus of the best underlying graph structure for modeling (Ortega et al., 2018; Scarselli et al., 2008; Ruiz et al., 2021). Often, this structure is unrelated to the graph signal itself (for example, geometric graphs for traffic signals). CoVariance Neural Networks (VNN) propose the use of the sample covariance matrix as the underlying graph shift operator (GSO)(Sihag et al., 2022). This approach encodes pairwise relationships in a robust statistical object. Yet, while this represents relevant interactions in a *static* case this does not necessarily hold when time-evolving graph signals are being modeled (Li and Zhu, 2016).

Graph temporal convolutional neural networks (GTCNN) (Isufi and Mazzola, 2021; Sabbaqi and Isufi, 2023) are a notable development in the spatio-temporal modeling of dynamically evolving graph signals. This class of models typically constructs a fully connected Cartesian or Kronecker product graph. While this effectively captures instantaneous interactions, convolutions in this domain result in a computational complexity that is quadratic in time, thus infeasible for longer time-series (Leskovec et al., 2010).

Given a time-evolving multi-variate signal the sample covariance represents the *long-term* correlation between variables over the entire time period. However, each snap shot in time has varying *instantaneous* interactions(Roy et al., 2024). This difference is in fact, non-trivial. While approaches like temporal PCA (Scharf et al., 2022) perform projections over the time averaged sample covariance matrix, this aggregation loses potentially useful information. This is demonstrated by the development of the time-varying graphical lasso (Hallac et al., 2017), an optimization framework that estimates a dynamic inverse covariance matrix directly from time series data. While this approach is intuitive and useful, the large computational cost of solving such an optimization problem has limited the use of this approach in neural network architectures (Hamilton et al., 2017).

Graph Variate Signal Analysis (GVSA)(Smith et al., 2019) provides an extended general framework to GSP for the analysis of spatio-temporal signals, using general instantaneous pairwise node functions (unrestricted by matrix multiplication) to formulate data constructed dynamic graph structures. This framework motivates methods such Graph-Variate Dynamic Connectivity and FAST Functional Connectivity, where these instantaneous graphs are filtered by a stable support constructed from the long term signal coupling information of the signal itself (GVDC) or a global cohort (FAST), reducing noise in short temporal windows while providing a very high, sample by sample, temporal resolution which does not rely on a window length compared to traditional sliding window approaches.

In this work we integrate GVSA with the more traditional "convolution" aggregation found in modern Graph Neural Networks (GNNs) (Li et al., 2016; Abadal et al., 2021; Pfrommer et al., 2021; Isufi et al., 2024; Veličković et al., 2018). For each input into the network an instantaneous connectivity tensor against a stable (and potentially learnt) support is constructed. This tensor is multiplied with its respective signal vector, this results in the capturing of spatio-temporal functional interactions. With this, we derive two important theoretical insights. Firstly, we show that while instantaneous connectivity matrices are typically rank-deficient and non-invertible, Hadamard multiplication with a full-rank stable support remedies this. Furthermore, we show that by using parallelized batch processing and low-rank matrix construction we achieve a speed up resulting in a linear time-complexity. This allows, for the first time, the capture of sample resolution signal dependent connectivity in a efficient, scalable manner.

We evaluate GVNN forecasting performance in 3 chaotic maps,2 weather forecasting tasks and 2 EEG motor imagery tasks. GVNNs successfully capture the non-trivial instantaneous temporal interactions present in multi-variable time-series. Particularly, we show that it outperforms the state of the art conventional graph based methods for time-series. Showing that the inductive bias provided by GVNNs improve performance. In application, we study EEG motor imagery classification, demonstrating that GVNNs capture the high temporal resolution of EEG signals while effectively reducing noise outperforming approaches such as EEGNet(Lawhern et al., 2016) and the Transformer model. Our results indicate that GVNNs could play a pivotal role in advancing the next generation of Brain–Computer Interfaces (BCIs)(Aristimunha et al., 2023; Keutayeva et al., 2024; Zhang and Liu, 2018), where minimizing calibration time and maximizing online responsiveness are crucial engineering challenges(Bessadok et al., 2021).

2 Background and Motivation

2.1 Graph Neural Networks

Graph Signal Processing (GSP) extends classical signal processing to data indexed by the vertices of a graph. A key component is the Graph Shift Operator (GSO), whose eigen-decomposition underlies operations analogous to the Discrete Fourier Transform (DFT). These components are the foundation on which Graph Neural Networks are built (Isufi et al., 2024; Maskey et al., 2023; Levie et al., 2020).

Definition 1 (Graph Convolutional Filter)**.** Let $\mathbf{h} = [h_0, \ldots, h_K]^\top$ be filter coefficients. A graph convolutional filter of order K is the linear map

$$\mathbf{H}(\mathbf{S})\,\mathbf{x} = \sum_{k=0}^{K} h_k\,\mathbf{S}^k\mathbf{x} = H(\mathbf{S})\,\mathbf{x}, \tag{1}$$

where $\mathbf{H}(\mathbf{S}) = \sum_{k=0}^{K} h_k\mathbf{S}^k$.

Definition 2 (Graph Fourier Transform (GFT))**.** For a diagonalizable GSO $\mathbf{S} = \ \mathbf{V}\Lambda\mathbf{V}^{-1}$ with eigenvectors V and eigenvalues Λ, the GFT of a graph signal $\mathbf{x}$ is $\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$, and the inverse GFT is $\mathbf{x}\ \ = \mathbf{V}\,\tilde{\mathbf{x}}$.

Definition 3 (Graph Convolutional Network (GCN))**.** A Graph Convolutional Network (Sandryhaila and Moura, 2013; Zügner and Günnemann, 2019; Keriven and Peyré, 2019; Hamilton et al., 2017) layer updates a graph signal $\mathbf{X} \in \mathbb{R}^{N\times F}$ (with N nodes and F input features) as:

$$\mathbf{X}^{(\ell+1)} = \sigma\Big(\mathbf{H}(\mathbf{S})\,\mathbf{X}^{(\ell)}\mathbf{W}^{(\ell)}\Big)\,, \tag{2}$$

where $\mathbf{S}$ is the graph shift operator (GSO) of choice, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{F_\ell\times F_{\ell+1}}$ are learnable weights, and $\sigma(\cdot)$ is a nonlinear activation function.

2.2 Graph-Time Convolutional Neural Networks (GTCNNs).

The standard way to model spatiotemporal signals is the use of product graphs to create Graph time Convolutional filters and thus GTCNN's.

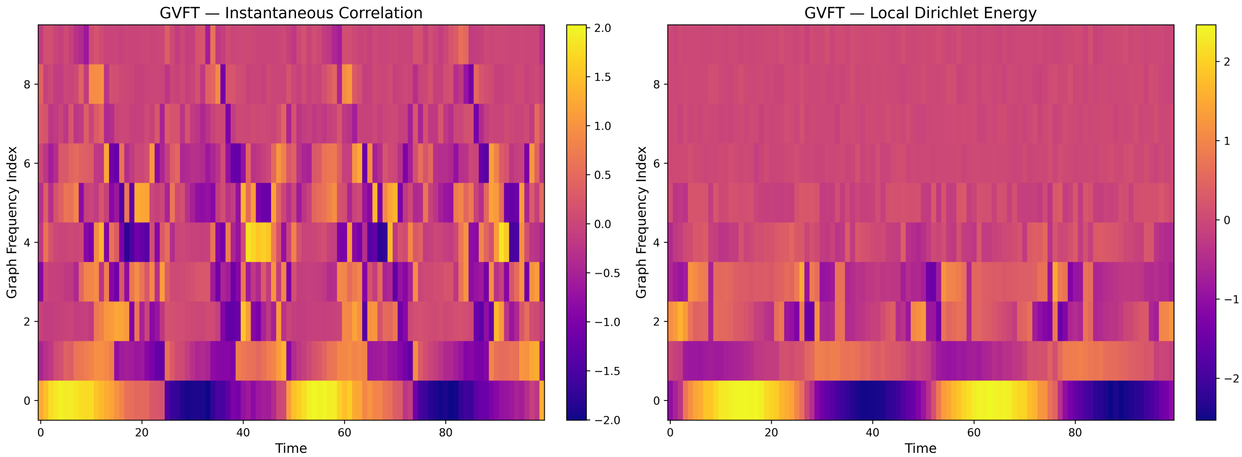

Figure 1: Graph Variate Fourier Transform (GVFT). Each panel shows the GVFT coefficients of a synthetic multivariate time series projected onto the eigenbasis of its own graph-structured connectivity profile at each time step. The left heatmap uses a squared-difference formulation for $\Omega_t = (x_i - x_j)^2 \cdot C$, while the right uses instantaneous correlation: $\Omega_t = \mathrm{corr}(x_t) \cdot C$, where C is the long-term correlation matrix across the full signal. The GVFT transforms the input signal $X \in \mathbb{R}^{N \times T}$ into a new matrix $\hat{X} \in \mathbb{R}^{N \times T}$, where each column represents the projection of x_t onto the eigenbasis of Ω_t. This figure illustrates how different formulations of signal-derived connectivity affect the spectral content and dynamics of the transformed signal.

Definition 4 (Graph-Time Convolutional Neural Network (GTCNN) (Isufi and Mazzola, 2021; Sabbaqi and Isufi, 2023))**.** Let $\mathcal{G}_{\mathcal{P}} = (\mathcal{V}_{\mathcal{P}}, \mathcal{E}_{\mathcal{P}}, \mathbf{S}_{\mathcal{P}})$ be a spatio-temporal product graph with shift operator $\mathbf{S}_{\mathcal{P}} \in \mathbb{R}^{NT \times NT}$. A spatio-temporal signal $\mathbf{X} \in \mathbb{R}^{N \times T}$ is vectorized as $\mathbf{x}_{\mathcal{P}} = \mathrm{vec}(\mathbf{X}) \in \mathbb{R}^{NT}$.

The *graph-time convolutional filter* of order K is defined as

$$\mathbf{y} = \left(\sum_{k=0}^{K} h_k \, \mathbf{S}_{\mathcal{P}}^{k} \right) \mathbf{x}_{\mathcal{P}} \;\equiv\; \mathbf{H}(\mathbf{S}_{\mathcal{P}}) \, \mathbf{x}_{\mathcal{P}}, \tag{3}$$

which aggregates information from K-hop spatio-temporal neighborhoods.

For multiple features, let $\mathbf{X}_{\mathcal{P}}^{(\ell-1)} \in \mathbb{R}^{NT \times F_{\ell-1}}$ denote the input at layer $\ell - 1$. We apply a bank of polynomial filters with coefficient matrices $\{\mathbf{H}_k^{(\ell)}\}_{k=0}^{K}$. The propagation rule of layer ℓ is

$$\mathbf{X}_{\mathcal{P}}^{(\ell)} = \sigma\!\left(\sum_{k=0}^{K} \mathbf{S}_{\mathcal{P}}^{k} \, \mathbf{X}_{\mathcal{P}}^{(\ell-1)} \mathbf{H}_k^{(\ell)} \right), \tag{4}$$

where $\mathbf{H}_k^{(\ell)} \in \mathbb{R}^{F_{\ell-1} \times F_\ell}$ are trainable filter coefficient matrices and $\sigma(\cdot)$ is a pointwise nonlinearity (e.g., ReLU).

A L-layer GTCNN is obtained by stacking such modules.

While effective for shorter temporal sequences the clear bottleneck here is the quadratic dependency in *both* the number of nodes and sequence length, this makes the modeling of long time-series unfeasible. Furthermore the product graphs do not capture instantaneous signal specfici dependencies and are usually a binary graph.

2.3 Graph Variate Signal Analysis

A potential issue with GSP based neural network architectures is that the relationship and relevance of the underlying graph structure to the signal is unclear and typically unchanging. There have been recent progress in addressing this in the form of CoVariance Neural Networks (VNN). Here the sample covariance matrix is used a GSO, giving us a natural interpretation of Graph Convolution that is inherently linked to Principal Component Analysis (PCA) (Maćkiewicz and Ratajczak, 1993).

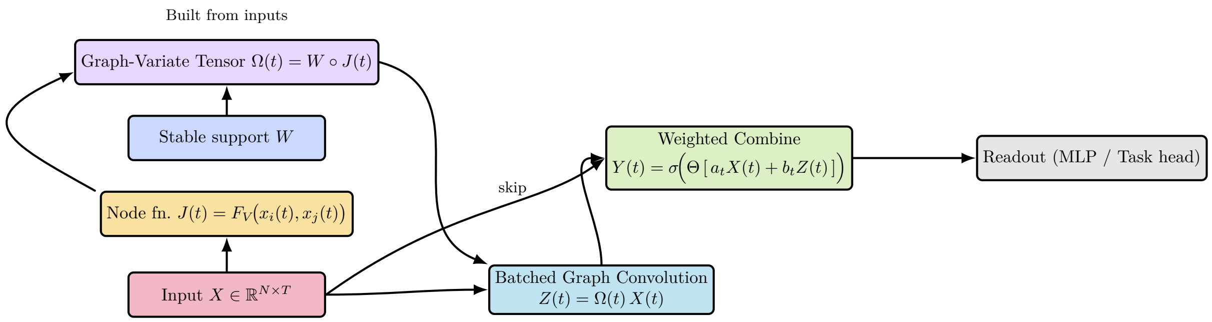
Built from inputs
Graph-Variate Tensor $\Omega(t) = W \circ J(t)$
Stable support W
Node fn. $J(t) = F_V\big(x_i(t), x_j(t)\big)$
Input $X \in \mathbb{R}^{N \times T}$
skip
Weighted Combine
$Y(t) = \sigma\Big(\Theta\,[\,a_t X(t) + b_t Z(t)\,]\Big)$
Batched Graph Convolution
$Z(t) = \Omega(t)\,X(t)$
Readout (MLP / Task head)

Figure 2: Graph-Variate Neural Network (GVNN) layer. A multivariate sequence $X \in \mathbb{R}^{N \times T}$ induces instantaneous connectivity $J(t)$, which is combined with a long-term support W to form $\Omega(t) = W \circ J(t)$. In parallel, X and $\Omega(t)$ drive a batched graph convolution $Z(t) = \Omega(t)X(t)$. A skip connection carries X to the combiner, which applies a learned linear map and a nonlinearity, $Y(t) = \sigma\big(\Theta[\,a_t X(t) + b_t Z(t)\,]\big)$.

Temporal data however, is dynamic (Manuca and Savit, 1996), i.e a single covariance estimation aggregating information over time may not be a suitable representation, particularly in the presence of irrelevant noise. Graph Variate Signal Analysis (GVSA) brings a sample-level, graph-weighted perspective to multivariate signals: it re-introduces node-to-node relationships in each time instant, but modulates their impact with a stable (or longer-term) graph. Importantly this *does not depend on a window length.* This yields time-varying connectivity estimates and graph metrics that are more robust against momentary noise yet still capture fine-grained transient dynamics. It has been shown that GVSA outperforms many sliding-window or purely instantaneous techniques (Smith et al., 2019).

Definition 5 (Graph-Variate Signal Analysis)**.** Let $\Gamma = (V, X, E, W)$ be a graph-variate signal, where

- V is the set of n nodes,
- $X \in \mathbb{R}^{n \times p}$ is the multivariate signal (each of the n nodes has p samples),
- E is the set of edges, and
- $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix with entries w_{ij}.

Define a bivariate *node-space function* F_V as

$$J_{ij}(t) \;=\; F_V\big(x_i(t),\, x_j(t)\big), \quad \text{for } i \neq j, \quad J_{ii}(t) = 0.$$

Graph-Variate Signal Analysis (GVSA) produces, at each time sample t, an $n \times n$ matrix given by the Hadamard (element wise) product

$$\Omega(t) \;=\; W \;\circ\; J(t),$$

whose entries are

$$\Omega_{ij}(t) \;=\; \big[W \circ J(t)\big]_{ij} \;=\; W_{ij}\, F_V\big(x_i(t),\, x_j(t)\big).$$

This, overall, gives a $N \times N \times T$ Tensor representation.

This framework not only allows a sample by sample high temporal resolution but is also computationally efficient. Note that no eigendecomposition is done at any stage and the entire analysis is in the node-space. Furthermore, node functions are typically chosen to exploit computational efficiency through low rank, vector outer product based operations. The stable support acts as an inherent stabilizer emphasizing stable long-term correlations and minimizing noise while still readily picking up instantaneous dynamics, providing a trade-off between global and local connectivity information. This is typically chose as the long-term correlation matrix of the signal itself or averaged over a cohort (Roy et al., 2024; Smith et al., 2019).

3 Graph Variate Neural Networks

By combining GSP and GVSA approaches we conjecture that time-step wise convolution of the graph signal with its own instantaneous temporal connectivity profile can exploit the rich spatio-temporal information present in many real-life signals.

In this vein, we define Graph-Variate Neural Networks as follows.

Definition 6 (Graph-Variate Neural Network (GVNN, layer-wise form))**.** Let $\mathbf{W} \in \mathbb{R}^{N \times N}$ be a stable (long-term) graph support. For an input sequence $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N \times T}$ at layer ℓ, denote its t-th column by $\mathbf{x}^{(\ell)}(t) \in \mathbb{R}^N$.

The input-dependent graph-variate tensor is

$$\Omega^{(\ell)}(\mathbf{X}^{(\ell)}) \in \mathbb{R}^{N \times N \times T}, \qquad \Omega_{ij}^{(\ell)}(t) = W_{ij}\, F_V\big(x_i^{(\ell)}(t),\, x_j^{(\ell)}(t)\big). \tag{5}$$

for a chosen bivariate function $F_V(\cdot, \cdot)$.

Let $\mathbf{a}^{(\ell)}, \mathbf{b}^{(\ell)} \in \mathbb{R}^T$ be learnable scalar filter coefficients (one per time step), and let $D_{\mathbf{a}^{(\ell)}} = \operatorname{diag}(\mathbf{a}^{(\ell)})$, $D_{\mathbf{b}^{(\ell)}} = \operatorname{diag}(\mathbf{b}^{(\ell)})$. Define the time-aligned multiplication

$$\big(\Omega^{(\ell)}(\mathbf{X}^{(\ell)}) * \mathbf{X}^{(\ell)}\big)_{:,t} \;=\; \Omega^{(\ell)}(t)\, \mathbf{x}^{(\ell)}(t), \qquad t = 1, \ldots, T. \tag{6}$$

Then the pre-activation output is

$$\mathbf{Z}^{(\ell)} \;=\; \mathbf{X}^{(\ell)} D_{\mathbf{a}^{(\ell)}} \;+\; \big(\Omega^{(\ell)}(\mathbf{X}^{(\ell)}) * \mathbf{X}^{(\ell)}\big)\, D_{\mathbf{b}^{(\ell)}}, \tag{7}$$

which is followed by a *trainable time-mixing weight block* $\Theta^{(\ell)} \in \mathbb{R}^{T \times T}$ and a pointwise activation $\sigma(\cdot)$:

$$\mathbf{X}^{(\ell+1)} \;=\; \sigma\big(\,\mathbf{Z}^{(\ell)}\Theta^{(\ell)}\,\big) \;\in\; \mathbb{R}^{N \times T}. \tag{8}$$

Stacking L such layers yields $\mathbf{X}^{(L)}$, which can be further mapped to a task-dependant readout layer.

Here, utilizing the fast batch based parallel processing allows a natural convolution operation where a spatio-temporal signal at a given timestep is convolved with its own connectivity profile. Also given the low rank nature of the connectivity profiles, we provide a robust platform to capture signal dependent functional inter-dependencies while being computationally efficient. Note also that we can optimize the stable support, and thus the entire dynamic connectivity profile, efficiently through training. This retains a high temporal resolution while allowing end-to-end optimization.

Equivalently, from a GSP lens, we can define the Graph-Variate Fourier Transform (GVFT) as projections of the signal vector onto its own temporal connectivity profile, this returns a matrix of size $N \times T$ that allows a simultaneous time-frequency decomposition. That is, each column of the GVFT represents the frequencies in terms of the eigenbasis of the functional graph at that time step.

Definition 7 (Graph Variate Fourier Transform)**.** Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$ be a spatio-temporal signal, where $\mathbf{x}_t \in \mathbb{R}^N$ is the t-th snapshot. For each t, define

$$\Omega_t = \big[f(x_t(i), x_t(j))\, W_{ij}\big]_{i,j=1}^{N}, \tag{9}$$

with $W \in \mathbb{R}^{N \times N}$ a connectivity matrix and $f(\cdot, \cdot)$ a symmetric node-pair function (e.g. $f(a, b) = (a - b)^2$). Since Ω_t is symmetric, it admits $\Omega_t = U_t \Lambda_t U_t^{\top}$. The GVFT of $\mathbf{x}_t$ is

$$\widehat{\mathbf{x}}_t = U_t^{\top} \mathbf{x}_t, \tag{10}$$

and stacking over time yields $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_T]$.

Definition 8 (Graph-Variate frequency response)**.** For a fixed time index t, let the instantaneous connectivity slice $\Omega(t) \in \mathbb{R}^{N \times N}$ be symmetric with eigendecomposition

$$\begin{aligned} \Omega(t) &= V_t \Lambda_t V_t^{\top}, \\ \Lambda_t &= \operatorname{diag}\big(\lambda_1(t), \ldots, \lambda_N(t)\big), \end{aligned}$$

Consider the *two-tap* Graph-Variate filter

$$y(t) \;=\; a_t\, x(t) + b_t\, \Omega(t)\, x(t), \qquad a_t, b_t \in \mathbb{R},$$

acting on an input vector $x(t) \in \mathbb{R}^N$. Define the Graph Fourier transforms

$$\tilde{x}(t) := V_t^{\top} x(t), \qquad \tilde{y}(t) := V_t^{\top} y(t).$$

Substituting the eigen-decomposition yields

$$\tilde{y}(t) \;=\; \big(a_t\, I_N + b_t\, \Lambda_t\big)\, \tilde{x}(t),$$

or component-wise,

$$\tilde{y}_i(t) \;=\; \underbrace{\big(a_t + b_t\, \lambda_i(t)\big)}_{h_t(\lambda_i(t))}\, \tilde{x}_i(t), \qquad i = 1, \ldots, N.$$

The scalar function

$$h_t(\lambda) := a_t + b_t\, \lambda \tag{11}$$

is called the *instantaneous frequency response* of the Graph-Variate filter at time t. Thus, spectrally, the filter acts as point-wise multiplication:

$$\tilde{y}_i(t) \;=\; h_t(\lambda_i(t))\, \tilde{x}_i(t). \tag{12}$$

Definition 6 is in direct analogy with the classical convolution theorem $\tilde{y}_i = \tilde{h}(\lambda_i)\, \tilde{x}_i$ for polynomial graph filters, but with a spectrum $\{\lambda_i(t)\}$ and a response h_t that are re-evaluated at every time step.

While we can clearly extend GVNNs by including higher order polynomials per time-step, we exclude these for the sake of simplicity. We further note that, computationally (and intuitively), a right multiplication with a time-wise filter coefficient matrix is more efficient then using polynomial filter coefficients (the typical choice in the GNN literature).

This dual perspective is a shift from the traditional GSP sense of graph frequencies given that the graph is constructed from the signal itself. In fact there is a closer link to PCA present here. Projecting signals onto a data-driven dynamic eigenbasis (i.e the sample Covariance matrix in PCA), supported by a stable support, allows a high level of precision and interpretability.

3.1 Temporal Signal Dependent Convolution

Temporal information provides rich, discriminative information that could significantly enhance machine learning models. As an example, EEG signals have a very high temporal resolution. While traditionally being studied in the frequency or spectral domain, the time domain analysis of EEG signals provide great potential in enhancing Brain Computer Interfaces (BCI).

We focus here on two common temporal domain node-space functions, given graph signals x we define:

- **Instantaneous correlation(IC):**

$$F_V(x_i(t), x_j(t)) = \left|\big(x_i(t) - \bar{x}_i\big)\, \big(x_j(t) - \bar{x}_j\big)\right|. \tag{13}$$

  where $\bar{x}_i = \frac{1}{T}\sum_{t=1}^{T} x_i(t)$ is the temporal mean of node i.
- **Local Dirichlet Energy (LDE) (Smith et al., 2017):**

$$F_V(x_i(t), x_j(t)) = \big(x_i(t) - x_j(t)\big)^2. \tag{14}$$

Instantaneous correlation is rank-1 and LDE rank-3, both expressible as sums of outer products. Such structures are efficient, as outer products reduce to parallelizable vector operations that GPUs compute rapidly. This approach combined with the Hadamard support are inspired from recent advances in Parameter–efficient fine-tuning (PEFT)(Hu et al., 2021; Huang et al., 2025), a method to improve the efficiency of Large Language Model's (LLM's). We direct the interested reader to A.11.

The instantaneous correlation captures co-deviation from mean temporal patterns.The LDE node function has a direct relationship to the Dirichlet energy and captures local node gradient changes. We can also take linear combinations of these two node functions in order to exploit both their contrasting views.

There is an important observation to make here with transformers here, given that the attention mechanism can be argued to use a "graph" constructed from the data itself for convolution (Vaswani et al., 2017), in fact recent ideas have provided a unifying view on Transformers and GNNs, arguing that transformers are GNN's that operate on a data-specific graph (Joshi, 2025). Thus GVNNs can be argued to be a form of attention with a fundamentally different formulation, i.e the time-step specific tensor weighted against a stable support. We expand on this in the appendix for the interested reader A.12.

Theorem 1 (Full-rank preservation under Hadamard filtering)**.** Let $J_{ij} = F_V(x_i(t), x_j(t)) = x_i(t)x_j(t)$ be the *unfiltered* instantaneous correlation profile with rank $m < N$. If every component of $\tilde{x}_t^{(m)}$ is non-zero, and W is of full rank then

$$\operatorname{rank}\big(\Omega(t) \;=\; W \;\circ\; J(t)\big) = N, \quad \text{i.e. } \Omega(t) \text{ is invertible.}$$

Moreover $\Omega(t)$ is symmetric positive-definite, preserving the signature of C.

Proof. See Appendix A.6 □

This theorem shows that Hadamard filtration with a stable support *induces* stability into the instantaneous correlation profile.

Figure 2 shows empirical evidence of Theorem 1 where the Hadamard filtered matrix by the full-rank long-term correlation matrix is now invertible and has a much lower condition number. We prove similar results for the LDE case in the appendix A.8.

The LDE connectivity profile has a distinct relationship with the traditional Dirichlet Energy of a signal (naturally encoding a measure of smoothness into signal convolutions) as shown in the following theorem.

Theorem 2 (Gershgorin–Dirichlet Bound)**.** Let $W \in \mathbb{R}^{N\times N}$ be symmetric and $x \in \mathbb{R}^N$ any signal. Form

$$J_{ij}(t) = F_V(x_i(t), x_j(t)) = (x_i(t) - x_j(t))^2, \quad \Omega(t) = W \circ J(t),$$

and define

$$\mathcal{E}_{\text{abs}} = \tfrac{1}{2}\sum_{i,j} |W_{ij}\,(x_i(t) - x_j(t))^2|.$$

the spectral radius is

$$\rho(\Omega(t)) \leq \; 2\,\mathcal{E}_{\text{abs}}(t)$$

Proof. See Appendix A.7 □

Theorem 2 shows that the spectral radius of the Hadamard Filtered LDE is upper bounded by twice the absolute Dirichlet energy of the signal on the stable support. Intuitively, this ensures that the GVNN convolution is *smoothness-aware* (See Appendix for more details). This relates the spectral radius of the LDE connectivity profile with the traditional Dirichlet Energy of a graph signal on the stable support W.

4 Experimental Results

4.1 Chaotic Maps

Chaotic systems pose unique challenges to statistical learning models and are also interpretable as benchmarks. They thus provide a baseline to compare GVNN's with other graph based models for time-series (Gilpin, 2023).

We compare GVNNs with a standard GTCNN, a Gated Graph RNN (GGRNN) and Graph VARMA (GVARMA) model. For our node function we used a linear combination of the local dirchlet energy and instantaneous correlation while allowing the stable support to be learnt from data.

We have chosen these models primarily due to their core operation being some sort of Graph Convolution. Note we are not considering hybrid models such as Graph Wavenet () however do foresee future work incorporating GVNNs into hybrid architectures. We have chosen these models primarily due to their core operation being some sort of Graph Convolution. Note we are not considering hybrid models such as Graph Wavenet (Wu et al., 2019) however we do foresee future work incorporating GVNNs into hybrid architectures.

For all models except GTCNN (Which uses the long-term correlation as the spatial component for fairness) we initialize the stable support with the long-term stable correlation of the chaotic map and let the model optimize this end-to-end. The node function was a linear combination of the LDE and instantaneous correlation.

We evaluate three multi-dimensional chaotic maps. The Coupled Lorenz, Hopfield and MacArthur maps

We see that GVNNs perform the best over all horizons on the Hopfield and Macarthur Map with large gains being visible in the MacArthur dataset in particular. In the coupled Lorenz map, while GVNNs perform the

Table 1: Chaotic datasets: MSE (↓) across horizons and time per epoch.

Dataset	Model	H=3	H=6	H=12	Time/epoch (s)
Hopfield	GVNN	**0.0237 ± 0.0008**	**0.1131 ± 0.0024**	**0.1837 ± 0.0053**	0.1
	GTCNN	0.1029 ± 0.0052	0.1683 ± 0.0014	0.2229 ± 0.0031	0.1
	GVARMA	0.5283 ± 0.0082	0.5846 ± 0.0086	0.6514 ± 0.0060	0.1
	GGRNN	0.0628 ± 0.0166	0.1742 ± 0.0083	0.2662 ± 0.0107	0.1
Lorenz	GVNN	**0.2143 ± 0.0083**	**0.5001 ± 0.1623**	0.7325 ± 0.0092	0.1
	GTCNN	0.8163 ± 0.0456	0.8595 ± 0.0282	0.9039 ± 0.0145	0.1
	GVARMA	0.8739 ± 0.0188	0.8764 ± 0.0397	0.9027 ± 0.0027	0.1
	GGRNN	0.3528 ± 0.0271	0.5327 ± 0.0159	**0.5971 ± 0.0049**	0.1
MacArthur	GVNN	**0.0910 ± 0.0004**	**0.2509 ± 0.0046**	**0.3914 ± 0.0087**	0.3
	GTCNN	0.8800 ± 0.0148	0.8479 ± 0.0123	0.8856 ± 0.0015	0.2
	GVARMA	0.5454 ± 0.0325	0.7608 ± 0.0794	0.8355 ± 0.0212	0.2
	GGRNN	0.2232 ± 0.0009	0.4252 ± 0.0099	0.5073 ± 0.0034	0.2

best over horizons of length 1 and 3, they are outperformed by GGRNNs over the horizon of length 5. This could be due to temporal interactions being less predictive for longer horizon in this chaotic map, further, a model incorporating a combination of GVNNs and GGRNNs may be promising.

4.2 Traffic Forecasting

We evaluate four graph-based forecasting models on the METR-LA and PEMS-BAY traffic networks. We also compare performance with the more commonly used transformer and LSTM models. METR-LA contains four months of speed measurements from 207 sensors in Los Angeles County at 5 minute intervals, and PEMS-BAY comprises six months of data from 325 sensors in the Bay Area at the same resolution(Sun et al., 2020; Li et al., 2018).

Following standard practice, we predict future speeds at horizons $h \in \{3, 6, 12\}$ time-steps (i.e. 15, 30, and 60 minutes ahead) given the past $T = 6$ observations (30 minutes). The graph based models follow the same layout as in the previous experiment. However, we evaluate the case of the two layer GVNN's with and without a trainable support W.

Table 2 shows our results. It can be noted that using a fixed support GVNN's outperform the graph based models but remain inferior to the LSTM and Transformer models. Allowing W to be learned however, results in large gains in performance where GVNN's significantly outperform all models. As these datasets have a large number of nodes we do observe GVNN's have a large increase in training time, however, we believe that the increase in performance justifies this decrease in speed.

Table 2: Final Test MSE (lower is better) for PEMS-BAY and METR-LA across all models.

Dataset	Model	Horizon 3	Horizon 6	Horizon 12	Time per epoch(s)
PEMS-BAY	GVNN (Trainable W)	**0.1722 ± 0.0093**	**0.2323 ± 0.0080**	**0.3250 ± 0.0229**	7.2
	Transformer	0.3126 ± 0.0099	0.3467 ± 0.0026	0.3858 ± 0.0061	1.1
	LSTM	0.3686 ± 0.0231	0.3810 ± 0.0085	0.4058 ± 0.0022	1.1
	GVNN(Static W)	0.7017 ± 0.0460	0.7642 ± 0.0611	0.8097 ± 0.0280	3.4
	GTCNN	0.9703 ± 0.0032	1.0010 ± 0.0099	1.0474 ± 0.0071	1.06
	GVARMA	0.7940 ± 0.0128	0.8271 ± 0.0113	0.8862 ± 0.0052	1.01
	GGRNN	0.8766 ± 0.0040	0.9175 ± 0.0061	0.9736 ± 0.0018	1.15
METR-LA	GVNN (Trainable W)	**0.2218 ± 0.0017**	**0.3082 ± 0.0158**	**0.4434 ± 0.0033**	2.4
	Transformer	0.2928 ± 0.0104	0.3799 ± 0.0072	0.5384 ± 0.0214	0.6
	LSTM	0.3554 ± 0.0054	0.4355 ± 0.0021	0.6644 ± 0.0280	0.6
	GVNN (Static W)	0.6012 ± 0.0625	0.6631 ± 0.0790	0.7076 ± 0.0301	1.1
	CPGraphST	0.9082 ± 0.0191	0.9234 ± 0.0211	0.9887 ± 0.0138	0.5
	GVARMA	0.9713 ± 0.0364	0.9527 ± 0.0447	1.0680 ± 0.0339	0.3
	GGRNN	0.8205 ± 0.0167	0.8621 ± 0.0089	0.9281 ± 0.0048	0.4

4.3 EEG Motor Imagery Tasks

The BCI Competition IV 2a dataset (Aristimunha et al., 2023) comprises EEG recordings from nine subjects performing four motor imagery tasks (left hand, right hand, feet, tongue) with data recorded in a 17 channel setup. The Physionet dataset comprises a dataset including EEG recordings of 109 healthy subjects. The participant imagines opening and closing their right or left fist and is a binary classification task. The data is recorded in a 64 channel setup.

We evaluate with cross fold validation using 5 independent data folds. For the BCI-2A dataset we use a fixed W set as the global long term correlation matrix computed from the training set and allow the W to be learned for the PhysioNet task.

Table 3: BCI-2A: Overall summary (K-fold CV)

Model	Accuracy (%)	Kappa	Time (s)
GVNN (LDE + Static W)	60.15 ± 1.21	0.4686 ± 0.0162	0.5
EEGNet	$\mathbf{60.51 \pm 3.88}$	$\mathbf{0.4735 \pm 0.0517}$	1.0
Transformer	51.99 ± 3.01	0.3598 ± 0.0401	1.5
LSTM	52.76 ± 2.27	0.3701 ± 0.0303	1.5

Table 4: PhysioNet: Overall summary (K-fold CV)

Model	Accuracy (%)	F1	Kappa	Time (s)
GraphVar+MLP (LDE + Learned W)	80.29 ± 0.82	0.8021 ± 0.0104	0.6058 ± 0.0164	2.0
Transformer	$\mathbf{80.94 \pm 0.87}$	$\mathbf{0.8095 \pm 0.0091}$	$\mathbf{0.6189 \pm 0.0173}$	0.9
LSTM	74.19 ± 1.74	0.7279 ± 0.0277	0.4834 ± 0.0351	1.4
EEGNet	79.61 ± 1.55	0.7959 ± 0.0145	0.5922 ± 0.0310	3.2

Table 3 and 4 show our results. As expected, GVNN's have a faster training speed on the lower channel BCI-2A dataset and is the fastest model with EEGNet only outperforming it slightly and significantly surpasses LSTM and Transformer models.

For the PhysioNet dataset we see an increase in training time for the GVNN model given the increase in channel count to 64 yet we still see competitive performance with the Transformer model while it outperforming EEGNet and being faster.

5 Conclusion and Limitations

In this work we have introduced Graph Variate Neural Networks- a general framework that constructs signal dependant dynamic graph structures in a computationally efficient manner by exploiting one-shot batch processing. We further introduced two interpretable node functions, the Local Dirichlet Energy and instantaneous correlation. We show theoretically how a stable support can 'stabilize' these low-rank instantaneous structures while also being computationally simple.

In the notoriously hard task of EEG motor imagery classification, we show that GVNNs are competitive with and sometimes outperform (in terms of efficiency) traditionally used models such as the Transformer Architecture or EEGNet. This improvement in performance was sustained in the forecasting of chaotic systems, where non-trivial instantaneous interactions are present. GVNNs retained their superiority in traffic forecasting tasks, strongly outperforming strong traditional and graph based baselines.

We note that while we effectively capture *intra*-channel connectivity, we are disregarding auto-correlative behaviour by not connecting nodes in the time dimension. However, the improvement in performance by including signal dependent graph structures and reduction in computational time justify this decision. Furthermore, mechanisms such as a temporal attention or convolutional module can be applied right after a GVNN layer to attend to inter time-dependencies.

We also note that our approach retains the quadratic complexity with the number of nodes such as in GTCNNs. This can become large when constructing signal specific connectivity profiles, however such an approach *would not be possible* using a product graph. Further work should also develop new node functions and stable supports, potentially incorporating spatial properties or even information theoretic measures.

Reproducibility Statement

All datasets used in this study are publicly available and open source. Detailed experimental settings, including model architectures, hyperparameters, and training procedures, are described in the main text and appendix. To facilitate reproducibility, the codebase implementing our methods will be made available upon reasonable request from the authors.

Ethics Statement

This work relies exclusively on open-source datasets that do not contain personally identifiable or sensitive information. We anticipate no direct harms, ethical concerns, or foreseeable negative societal impacts arising from this research. The proposed methods are intended for advancing scientific understanding and improving model efficiency in a responsible manner.

Large Language Model (LLM) Usage Statement

During the preparation of this manuscript, we made limited use of a large language model for two purposes: (i) assisting in code ideation and refactoring for clarity and efficiency, and (ii) tidying up the exposition of the text for grammar and readability. The core research ideas, experimental design,theory, implementation, methodology and validation are entirely the work of the authors. No parts of the manuscript were generated verbatim by the LLM, and all content was critically reviewed and edited by the authors prior to submission.

References

Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys*, 54(9):1–38, 2021.

Bernardo Aristimunha, Igor Carrara, Pierre Guetschel, Stefan Sedlar, Pedro Rodrigues, Jakub Sosulski, Dilin Narayanan, Erik Bjareholt, Bastien Quentin, Robin Tibor Schirrmeister, Emmanuel Kalunga, Laurent Darmet, Clement Gregoire, Asif Abdul Hussain, Raffaele Gatti, Vladyslav Goncharenko, Jörg Thielen, Thomas Moreau, Yannick Roy, Vijay Jayaram, Alexandre Barachant, and Sylvain Chevallier. Mother of all bci benchmarks (moabb), 2023.

Alaa Bessadok, Mohamed Ali Mahjoub, and Islem Rekik. Graph neural networks in network neuroscience, 2021. arXiv preprint arXiv:2106.03535.

Guido Dornhege, José del R. Millán, Thilo Hinterberger, Dennis J. McFarland, and Klaus-Robert Müller. *BCI2000: A General-Purpose Software Platform for BCI*, pages 359–368. 2007.

William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling, 2023. URL https://arxiv.org/abs/2110.05266.

David Hallac, Youngsuk Park, Stephen Boyd, and Jure Leskovec. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 205–213, 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications, 2017. arXiv:1709.05584.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. HiRA: Parameter-efficient hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=TwJrTz9cRS.

Elvin Isufi and Gabriele Mazzola. Graph-time convolutional neural networks. In *IEEE Data Science and Learning Workshop (DSLW)*, 2021.

Elvin Isufi, Fernando Gama, David I. Shuman, and Santiago Segarra. Graph filters for signal processing and machine learning on graphs. *IEEE Transactions on Signal Processing*, 2024. doi: 10.1109/TSP.2024.3349788.

Chaitanya K. Joshi. Transformers are graph neural networks, 2025. URL https://arxiv.org/abs/2506.22084.

Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of polynomial spectral graph filters. In *ICASSP*, pages 5350–5354, 2020.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. In *NeurIPS*, pages 7092–7101, 2019.

Aruzhan Keutayeva, Nikita Fakhrutdinov, and Bekzod Abibullaev. Compact convolutional transformer for subject-independent motor imagery eeg-based bcis. *Scientific Reports*, 14:25775, 2024. doi: 10.1038/s41598-024-73755-4.

Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou Po Hung, and Brent Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15, 2016.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

Ron Levie, Wen Huang, Luca Bucci, Michael Bronstein, and Gitta Kutyniok. Transferability of spectral graph convolutional neural networks, 2020. arXiv:1907.12972.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018. URL https://arxiv.org/abs/1707.01926.

Yang Li and Zhengyuan Zhu. Modeling nonstationary covariance function with convolution on sphere. *Computational Statistics & Data Analysis*, 104:233–246, 2016. ISSN 0167-9473. doi: 10.1016/j.csda.2016.07.001.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Radu Manuca and Robert Savit. Stationarity and nonstationarity in time series analysis. *Physica D: Nonlinear Phenomena*, 99(2):134–161, 1996. ISSN 0167-2789. doi: https://doi.org/10.1016/S0167-2789(96)00139-X. URL https://www.sciencedirect.com/science/article/pii/S016727899600139X.

Saurabh Maskey, Ron Levie, and Gitta Kutyniok. Transferability of graph neural networks: An extended graphon approach. *Applied and Computational Harmonic Analysis*, 63:48–83, 2023.

Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342, 1993. ISSN 0098-3004. doi: https://doi.org/10.1016/0098-3004(93)90090-R. URL https://www.sciencedirect.com/science/article/pii/009830049390090R.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, May 2018. doi: 10.1109/JPROC.2018.2820126.

Samuel Pfrommer, Alejandro Ribeiro, and Fernando Gama. Discriminability of single-layer graph neural networks. In *ICASSP*, pages 8508–8512, 2021. doi: 10.1109/ICASSP39728.2021.9414583.

Om Roy, Yashar Moshfeghi, Agustín Ibáñez, Francisco Lopera, Mario A. Parra, and Keith M. Smith. Fast functional connectivity implicates p300 connectivity in working memory deficits in alzheimer's disease. *Network Neuroscience*, 8(4):1467–1490, 2024.

Luana Ruiz, Fernando Gama, and Alejandro Ribeiro. Graph neural networks: Architectures, stability, and transferability. *Proceedings of the IEEE*, 109(5):660–682, 2021.

Mohammad Sabbaqi and Elvin Isufi. Graph-time convolutional neural networks: Architecture and theoretical analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Aliaksei Sandryhaila and José M. F. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Florian Scharf, Andreas Widmann, Carolina Bonmassar, and Nicole Wetzel. A tutorial on the use of temporal principal component analysis in developmental erp research – opportunities and challenges. *Developmental Cognitive Neuroscience*, 54:101072, 2022. doi: 10.1016/j.dcn.2022.101072.

Saurabh Sihag, Gonzalo Mateos, Corey McMillan, and Alejandro Ribeiro. Covariance neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, New Orleans, LA, 2022. Curran Associates.

Keith Smith, Benjamin Ricaud, Nauman Shahid, Stephen Rhodes, John M. Starr, Agustin Ibáñez, Mario A. Parra, Javier Escudero, and Pierre Vandergheynst. Locating temporal functional dynamics of visual short-term memory binding using graph modular dirichlet energy. *Scientific Reports*, 7:1–12, February 2017. ISSN 2045-2322. doi: 10.1038/srep42013.

Keith Smith, Loukas Spyrou, and Javier Escudero. Graph-variate signal analysis. *IEEE Transactions on Signal Processing*, 67(2):293–305, 2019. doi: 10.1109/TSP.2018.2881658.

Yizhou Sun, Yulong Wang, Kun Fu, Zhizhong Wang, Chuan Zhang, and Jieping Ye. Constructing geographic and long-term temporal graph for traffic forecasting, 2020. arXiv:2004.10958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Petar Veličković, Guillem Cucurull, Arantón Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling, 2019. arXiv:1906.00121.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Qiong Zhang and Yang Liu. Improving brain computer interface performance by data augmentation with conditional deep convolutional generative adversarial networks, 2018. arXiv:1806.07108.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *KDD*, pages 246–256, 2019.

A Appendix / supplemental material

A.1 Hardware

All experiments were run on a single **NVIDIA A100** GPU (40 GB VRAM). Training used **FP32** precision (no mixed precision), and all runs were executed on a single device without model or data parallelism.

A.2 Experimental Details: Chaotic Maps

We consider three standard discrete-time chaotic benchmarks(Gilpin, 2023): *Coupled Lorenz*: a network of Lorenz oscillators with diffusive coupling between state variables, producing high-dimensional, synchronized–desynchronized regimes; *Hopfield map*: a discrete-time Hopfield network with frustrated connectivity (competing attractors) yielding complex transient dynamics; *MacArthur map*: a discrete-time ecological competition model (species competing for shared resources) exhibiting multi-species chaotic population fluctuations. Each dataset provides multivariate sequences $X \in \mathbb{R}^{N \times T}$ (channels = N nodes).

We use a sliding window of length $T{=}3$ to forecast horizons $H \in \{1, 3, 5\}$ (one-, three-, and five-step ahead). Windows slide with stride 1. Data are split chronologically into 80% train+val and 20% test; within the first

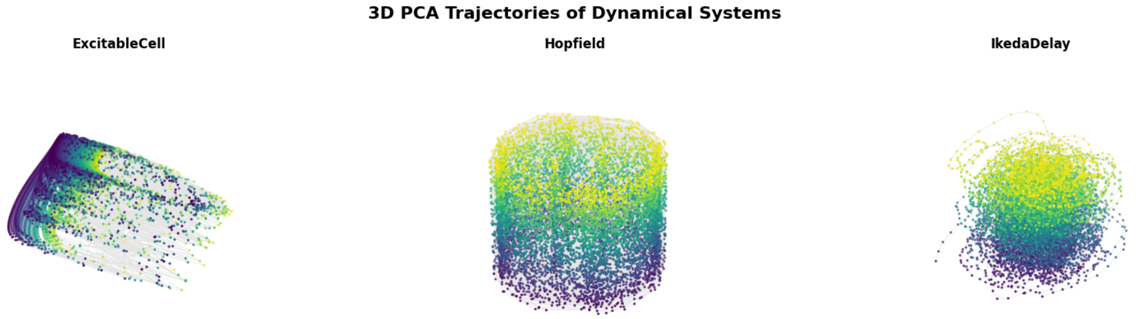

Figure 3: PCA of Chaotic Maps

Table 5: Chaotic maps forecasting: dataset-level hyperparameters (identical across maps).

Map	T	H	Stride	Split	Batch	Epochs	Seeds	Norm
Coupled Lorenz	3	1,3,5	1	80/20 (chron.)	128	500	124, 14, 124235	per-sample z-score (channels)
Hopfield	3	1,3,5	1	80/20 (chron.)	128	500	124, 14, 124235	per-sample z-score (channels)
MacArthur	3	1,3,5	1	80/20 (chron.)	128	500	124, 14, 124235	per-sample z-score (channels)

80% we take 80% train and 20% validation. Inputs are z-scored per sample across channels. All graph-based models use a *trainable* support W_C initialized from the long-term channel wise Pearson correlation over the *training* split , and fuse instantaneous operators by Hadamard product; dynamic slices are re-normalized as $D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}$. All models use 1 convolution layer with the GVNN using a linear combination of the two node functions. We treat GTCNN as a simple baseline with it's spatial component being the **fixed** long-term correlation matrix and the rest of the models allow end to end training of the graph. All models consist of a MLP readout layer with Leaky ReLU activation.

We train for 500 epochs with Adam (MSE loss), batch size 128, and report the best-validation checkpoint on test. Unless otherwise stated, we use three seeds $\{124, 14, 124235\}$ and hidden dimension 128.

A.3 Experimental Details: Traffic Forecasting (METR–LA & PEMS–BAY)

We use a sliding window of $T{=}6$ (30 min) to forecast $H \in \{3, 6, 12\}$ steps (15/30/60 min). Data are split chronologically: 80% train+val and 20% test; within the first 80% we take 80% train and 20% validation. Inputs are z-scored *per sample across channels*. Dynamic adjacencies are renormalized slice-wise as $D^{-\frac{1}{2}}(A+I)D^{-\frac{1}{2}}$. We train with Adam and MSE loss for 200 epochs, select the best validation checkpoint, and evaluate on test. Runs use three seeds $\{124, 14, 124235\}$. All models use 2 convolution layers with the GVNN having the LDE as the first layer and IC as second. The transformer and LSTM models also use 2 layers with the transformer only consisting of one attention head.

Table 7: Dataset-level hyperparameters (both model families run on both datasets; the *only* dataset difference is batch size).

Dataset	T	H	Split	Batch	Epochs	Seeds	Optimizer / Loss
METR–LA	6	3,6,12	80/20 (chron.)	1280	200	124, 14, 124235	Adam / MSE
PEMS–BAY	6	3,6,12	80/20 (chron.)	1024	200	124, 14, 124235	Adam / MSE

Graph construction (used by all graph-based models). We build a static channel similarity matrix $W_C \in \mathbb{R}^{C\times C}$ from channelwise Pearson correlations over the full training set . Models that mark W_C as *trainable* initialize from this correlation and update it end-to-end; otherwise W_C is fixed. All dynamic operators $\Omega(t)$ are fused with W_C by Hadamard product and renormalized slice-wise.

Table 6: Model hyperparameters and operator details for chaotic maps (all are graph-based; W_C is *trainable* and initialized from long-term correlation).

Model	**LR**	**Hidden**	**Epochs**	**Trainable W_C**
GTCNN	1×10^{-4}	128	500	Yes
GVARMA (P=1, Q=1, K=2)	1×10^{-4}	128	500	Yes
GGRNN	1×10^{-4}	128	500	Yes
GVNN	1×10^{-4}	128	500	Yes

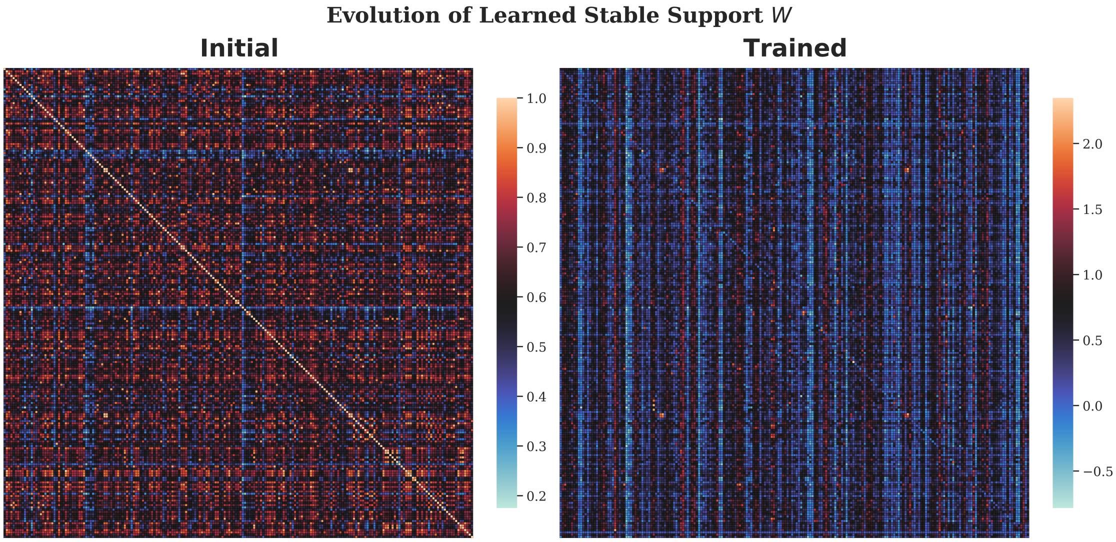

Figure 4: Learned graph support matrix W_C before and after training. The figure illustrates how the static graph support matrix W_C evolves through training. The left panel shows the initialized matrix, while the right panel presents the learned weights after optimization, revealing how the model adapts graph connectivity structure for improved forecasting.

Table 8: Model hyperparameters and operator details (applied identically on METR–LA and PEMS–BAY).

Family	**Model**	**LR**	**Hidden**	**Epochs**	**Trainable W_C**
Graph-based	GVNN	1×10^{-4}	128	200	No
	GTCNN	1×10^{-4}	128	200	No
	GVARMA (P=1, Q=1, K=2)	1×10^{-4}	128	200	No
	GGRNN	1×10^{-4}	128	200	No
	GVNN	1×10^{-4}	128	200	Yes
Sequence-based	LSTM (2 layers)	1×10^{-3}	128	200	—
	Transformer (1 head, 2 layers)	1×10^{-3}	128	200	—

A.4 EEG Experiments: Datasets and Hyperparameters

[1]

PhysioNet MI (binary: T1 vs. T2). We load raw EDF files from per-participant folders `S{001..109}` =, excluding faulty IDs $\{088, 089, 092, 100\}$. For each valid subject we select only the motor imagery runs

[1] For both EEG datasets the Transformer and LSTM models consisted of two layers while the GVNN was 1 layer.

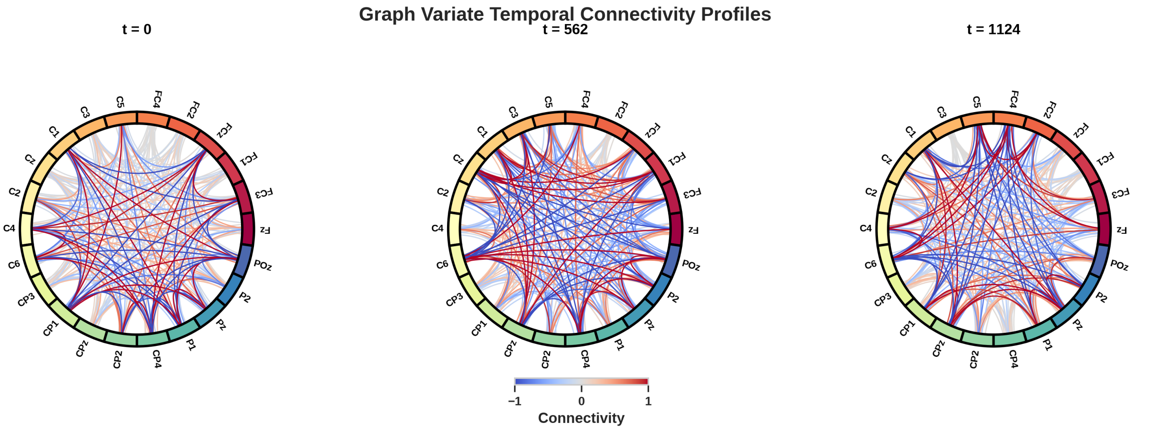

Figure 5: Instantaneous Correlation Connectivity profiles in the BCI-2A Multivariate Time series.

`R04`, `R08`, `R12`, read EDF with `mne`, and extract events from annotations. We dynamically map the annotation codes for `T1` and `T2`, keep only those trials, and epoch each trial with $t_{\min}$=0 to $t_{\max}$=3.1 s at 160 Hz (496 samples). Trials and labels are concatenated across all participants. We then perform stratified K=5-fold CV across *all* trials (pooled cross-subject), building the stable support W_C *within each fold from training windows only* as the absolute channelwise Pearson correlation |corr| . All models receive inputs normalized per sample across channels (z-score), and all graph operators use slice-wise symmetric renormalization $D^{-1/2}(A+I)D^{-1/2}$.

Table 9: PhysioNet MI: dataset-level protocol and hyperparameters.

Trials	Classes	Epoch	FS	CV	Batch	Epochs	LR / WD	Metrics
pooled (all subj.)	2 (T1/T2)	3.1 s (T=496)	160 Hz	strat. 5-fold	64	50	10^{-3} / 10^{-4}	Acc

BNCI2014_001 (BCI 2a, 4-class). We use MOABB/Braindecode (Aristimunha et al., 2023) to load all subjects (1..9). Preprocessing: pick EEG, scale by 10^6, band-pass $0.01-20$ Hz, exponential moving standardization (`factor_new`=10^{-3}, `init_block_size`=1000). Windows are created from events with a start offset of -0.5 s (MOABB defaults for stop/length are used). We concatenate windows across subjects and run stratified 5-fold CV. In each fold, W_C = |corr| is computed from training windows only, and used by GVNN; inputs are per-sample channel z-scored inside each model (Dornhege et al., 2007).

Table 10: BNCI2014_001 (4-class): dataset-level protocol and hyperparameters.

Trials	Classes	Preproc	CV	Batch	Epochs	LR / WD	Metrics
pooled (all subj.)	4	bp. $0.01-20$ Hz & EMS	strat. 5-fold	64	100	10^{-3} / 10^{-4}	Acc

A.5 Computational Complexity Analysis

We compare two *hypothetical* ways to realize signal-dependent graph convolution on inputs $x \in \mathbb{R}^{B\times C\times T}$, with a fixed spatial support $W_s \in \mathbb{R}^{C\times C}$ and temporal path adjacency $L_T \in \mathbb{R}^{T\times T}$.

1. **Naive Cartesian (Kronecker) Method.** For each sample b, compute per-time masked connectivity and then build a full spatiotemporal kernel by the Kronecker product with L_T, yielding $K_b \in \mathbb{R}^{(CT)\times(CT)}$, and apply K_b to $\hat{x}_b$.
2. **Proposed Graph-Variate Low-Rank Batched Method.** Construct rank-1 (`IC`) or *rank-3 expanded quadratic* (`LDE`) connectivities on-the-fly, mask by W_s via Hadamard product, and perform T batched mat–vecs *without* explicit Kronecker expansion.

1. Naive Product Graph Method

For each $b = 1, \dots, B$ and $t = 1, \dots, T$:

(a) *Per-time connectivity & masking*

$$S^{\mathrm{IC}}_{b,t} = x_{b,:,t}\, x_{b,:,t}^{\top}, \qquad S^{\mathrm{LDE}}_{b,t} = (x_{b,:,t} \odot x_{b,:,t})\, \mathbf{1}^{\top} + \mathbf{1}\, (x_{b,:,t} \odot x_{b,:,t})^{\top} - 2\, x_{b,:,t}\, x_{b,:,t}^{\top}, \qquad \widetilde{S}_{b,t} = S_{b,t} \circ W_s.$$

(b) *Stacking* $\widetilde{S}_b = [\widetilde{S}_{b,1}, \dots, \widetilde{S}_{b,T}] \in \mathbb{R}^{C \times C \times T}$.

(c) *Kronecker expansion & apply* $K_b = L_T \otimes \widetilde{S}_b \in \mathbb{R}^{(CT) \times (CT)}$, $\hat{y}_b = K_b\, \hat{x}_b$.

Complexity. Per-time connectivity: $O(BC^2T)$; kernel formation: $O(BC^2T^2)$; application: $O(BC^2T^2)$; memory for all K_b: $O(BC^2T^2)$. Net time: $O(BC^2T^2)$; memory: $O(BC^2T^2)$.

2. Proposed Graph-Variate Low-Rank Batched Method

Form, for all (b, t) in parallel,

$$J^{\mathrm{IC}}_{b,t} = x_{b,:,t}\, x_{b,:,t}^{\top}, \qquad J^{\mathrm{LDE}}_{b,t} = (x_{b,:,t} \odot x_{b,:,t})\, \mathbf{1}^{\top} + \mathbf{1}\, (x_{b,:,t} \odot x_{b,:,t})^{\top} - 2\, x_{b,:,t}\, x_{b,:,t}^{\top},$$

then mask with W_s: $\Omega_{b,t} = J_{b,t} \circ W_s$ (using the appropriate case). All T masked matrices live implicitly inside $\Omega \in \mathbb{R}^{B \times C \times C \times T}$. We then compute, in one batched call,

$$y_{b,:,t} = \Omega_{b,t}\, x_{b,:,t},$$

vectorizing over b and t.

Complexity. Connectivity+masking: $O(BC^2T)$; T batched mat–vecs: $O(BC^2T)$; memory $O(BC^2T)$. Net time: $O(BC^2T)$; memory: $O(BC^2T)$.

Table 11: Asymptotic comparison: naive Cartesian vs. batched low-rank.

Method	Time	Memory
Naive Cartesian (Kronecker)	$O(BC^2T^2)$	$O(BC^2T^2)$
Batched Low-Rank (ours)	$O(BC^2T)$	$O(BC^2T)$

> **Takeaway.** Avoiding explicit Kronecker formation with L_T removes the quadratic dependence on T in both compute and memory. Using rank-1 (`IC`) and rank-3 expanded quadratic (`LDE`) constructions, plus Hadamard masking and batched mat–vecs, yields linear $O(BC^2T)$ execution.

Main Convolution

Listing 1: Core PyTorch implementation of normalization, graph construction, and convolution.

```python
import torch

EPS = 1e-5

def renormalize_dynamic(A, eps=EPS):
    """
    A: (B, C, C, T) dynamic affinity
    Returns symmetric renorm:  D^{-1/2} (A + I) D^{-1/2}
    """
    I = torch.eye(A.size(1), device=A.device)[None, :, :, None]   # (1, C, C, 1)
    At = A + I
    deg = At.sum(2, keepdim=True)                                  # (B, C, 1, T)
    inv = deg.clamp(min=eps).pow(-0.5)
    S = inv * At * inv.transpose(1, 2)                             # symmetric
        renorm
    return S

def graph_variate(x, fun=’corr’, Zave=True, eps=EPS):
    """
    x: (B, C, T)
    returns normalized dynamic adjacency Om: (B, C, C, T)
    """
    B, C_, T_ = x.shape
    if Zave:
        mu = x.mean(1, keepdim=True)
        sig = x.std(1, keepdim=True, unbiased=True)
        x = (x - mu) / (sig + eps)

    if fun == ’sqd’:
        D  = x - x.mean(1, keepdim=True)
        Om = (x.unsqueeze(2) - x.unsqueeze(1)).pow(2)
    elif fun == ’corr’:
        D  = x - x.mean(2, keepdim=True)              # zero-mean over time
        Om = D.unsqueeze(2) * D.unsqueeze(1)          # rank-1 outer per time


    return Om

def graph_conv(x, Om):
    """
    x:  (B, C, T)
    Om: (B, C, C, T) dynamic (optionally renormalized) adjacency
    returns: (B, C, T)
    """
    Om_t  = Om.permute(0, 3, 1, 2)              # (B, T, C, C)
    sig_t = x.permute(0, 2, 1).unsqueeze(-1)   # (B, T, C, 1)
    out   = torch.matmul(Om_t, sig_t).squeeze(-1)  # (B, T, C)
    return out.permute(0, 2, 1)
```

In practice, we build Ω via `graph_variate`, apply the spatial mask (Hadamard with W_s), optionally call `renormalize_dynamic` slice-wise, and then use `graph_conv` to perform all BT mat–vecs in one call—achieving $O(BC^2T)$ time and memory.

A.6 Proof of Theorem 1

We first introduce the following defintions **Definition 1.** we observe T time-centered samples $x_t \in \mathbb{R}^N$ for $t = 1, \dots, T$, and define

$$\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t, \quad \tilde{x}_t = x_t - \bar{x}$$

and we assume

$$\boxed{W \succ 0} \quad \text{(PSD)}.$$

Let $w_{ij} = W_{ij}$. For each fixed t, define the stabilized *instantaneous correlation profile*

$$\rho_t(i,j) = W_{ij}\,\big|\tilde{x}_i^{(m)}(t)\,\tilde{x}_j^{(m)}(t)\big|, \quad i,j = 1,\dots,N.$$

Definition 2 (Sylvester's Law of Inertia). Let $A \in \mathbb{S}^N$ be a symmetric matrix of rank r with inertia $(p, q, 0)$, meaning p positive and q negative eigenvalues such that $p + q = r$. Then A is congruent to the diagonal normal form

$$G = \begin{pmatrix} I_p & 0 & 0 \\ 0 & -I_q & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad p + q = r.$$

Two symmetric matrices are congruent if and only if they have the same rank and signature $(p, q, 0)$.

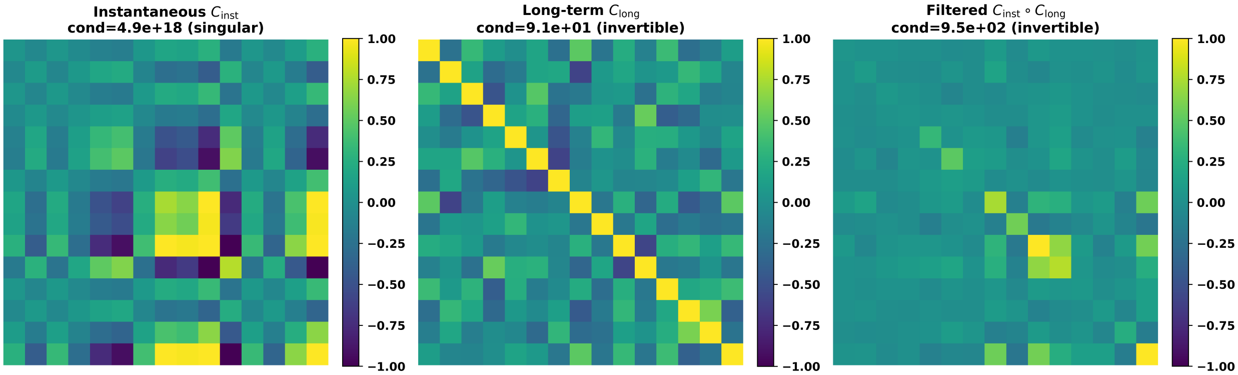

Figure 6: Comparison of instantaneous correlation profile, long-term covariance, and Hadamard-filtered covariance matrices. Each panel displays the respective matrix with its condition number and invertibility status.

Proof. of Theorem 1

Set

$$d_t := \big|\tilde{x}_t\big| \in \mathbb{R}^N, \qquad D_t := \operatorname{diag}\big(d_t\big),$$

so each $D_t^{(m)}$ is diagonal with strictly positive entries and thus invertible. The Hadamard product identity gives

$$\Omega(t) = W \circ \big(\tilde{x}_t \tilde{x}_t^{(m)\top}\big) = D_t\, W\, D_t,$$

i.e. $\Omega(t)$ is congruent to C.
Now applying Sylvester's Law, since $C \succ 0$ has inertia $(N, 0, 0)$, any matrix congruent to it must share the same inertia. Therefore

$$\Omega(t) \succ 0, \quad \operatorname{rank}(\Omega(t)) = \operatorname{rank}(W) = N.$$

This establishes both invertibility and positive-definiteness.

This completes the proof. □

A.7 Proof of Theorem 2

Proof. Recall Gershgorin's circle theorem: if $A = (a_{ij})$ is any $N \times N$ matrix then each eigenvalue λ of A satisfies

$$\lambda \in D\big(a_{ii},\, R_i(A)\big) \quad \text{where} \quad R_i(A) = \sum_{j \neq i} |a_{ij}|.$$

In our case $\Omega(t)_{ii} = 0$ and

$$R_i(\Omega(t)) = \sum_{j \neq i} |\Omega(t)_{ij}| = \sum_{j \neq i} |W_{ij}\, (x_i(t) - x_j(t))^2|,$$

so every eigenvalue δ of Ω lies in one of the real intervals $[-R_i, R_i]$. Taking the union over i gives

$$\rho(\Omega) \subset \bigcup_{i=1}^{N} [-R_i, R_i] = \big[-\max_i R_i,\ \max_i R_i\big].$$

By definition,

$$R_i = \sum_{j \neq i} |W_{ij}\, (x_i(t) - x_j(t))^2|.$$

Summing these radii over all i yields

$$\sum_{i=1}^{N} R_i = \sum_{i=1}^{N} \sum_{j\neq i} \big| W_{ij}\,(x_i(t) - x_j(t))^2 \big| = \sum_{i,j} \big| W_{ij}\,(x_i(t) - x_j(t))^2 \big| = 2\,\mathcal{E}_{\text{abs}}.$$

Thus the total "Gershgorin mass" equals twice the Dirichlet energy.

Since $\rho(\Omega) = \max|\delta| \leq \max_i R_i$, we need only show $\max_i R_i \leq 2\mathcal{E}_{\text{abs}}$. But from Step 2, $\sum_i R_i = 2\mathcal{E}_{\text{abs}}$, and the largest term in a sum of nonnegative numbers is no bigger than the sum itself. Hence

$$\max_i R_i \;\leq\; \sum_i R_i = 2\,\mathcal{E}_{\text{abs}},$$

hence

$$\rho(\Omega) \leq 2\,\mathcal{E}_{\text{abs}}.$$

completing the proof. □

A.8 Rank-Lifting of the LDE Connectivity profile

Theorem 3. Let $x_1, \ldots, x_N$ be N distinct real numbers and define the instantaneous squared-difference matrix

$$J(t) \in \mathbb{R}^{N\times N}, \qquad J_{ij}(t) = (x_i - x_j)^2,\; J_{ii}(t) = 0.$$

Let

$$\mathcal{W} = \{\, C \in \mathbb{R}^{N\times N} : C_{ij} \neq 0 \text{ for all } i \neq j\},$$

and for each $C \in \mathcal{W}$ form the Hadamard product

$$\Omega(t) = J(t) \circ W, \qquad \Omega_{ij} = J_{ij}(t)\,W_{ij}.$$

Then:

1. $\operatorname{rank}(D) \leq 3$, hence $\det(J(t)) = 0$ and D is singular.
2. The determinant
$$P(C_{12}, C_{13}, \ldots, C_{N-1,N}) \;=\; \det\big(D \circ C\big)$$
is a nonzero polynomial in the off-diagonal entries of C. Consequently, outside its algebraic zero-locus of Lebesgue measure 0, one has
$$\det(J(t) \circ W) \neq 0, \quad \operatorname{rank}(J(t) \circ W) = N,$$
so the Hadamard-weighted matrix is generically invertible.

Proof. **(1)** $\operatorname{rank}(J(t)) \leq 3$**.** Define column-vectors in $\mathbb{R}^N$ by

$$u_i = x_i^2(t), \quad v_i = x_i(t), \quad 1_i = 1.$$

Then

$$J_{ij}(t) = (x_i(t) - x_j(t))^2 = u_i\,1_j - 2\,v_i\,v_j + 1_i\,u_j,$$

so in matrix form

$$J(t) = u\,1^T \;-\; 2\,v\,v^T \;+\; 1\,u^T.$$

Each term on the right is rank 1, hence $\operatorname{rank}(J(t)) \leq 1 + 1 + 1 = 3$. In particular when $N > 3$, $J(t)$ is singular and $\det(J(t)) = 0$.

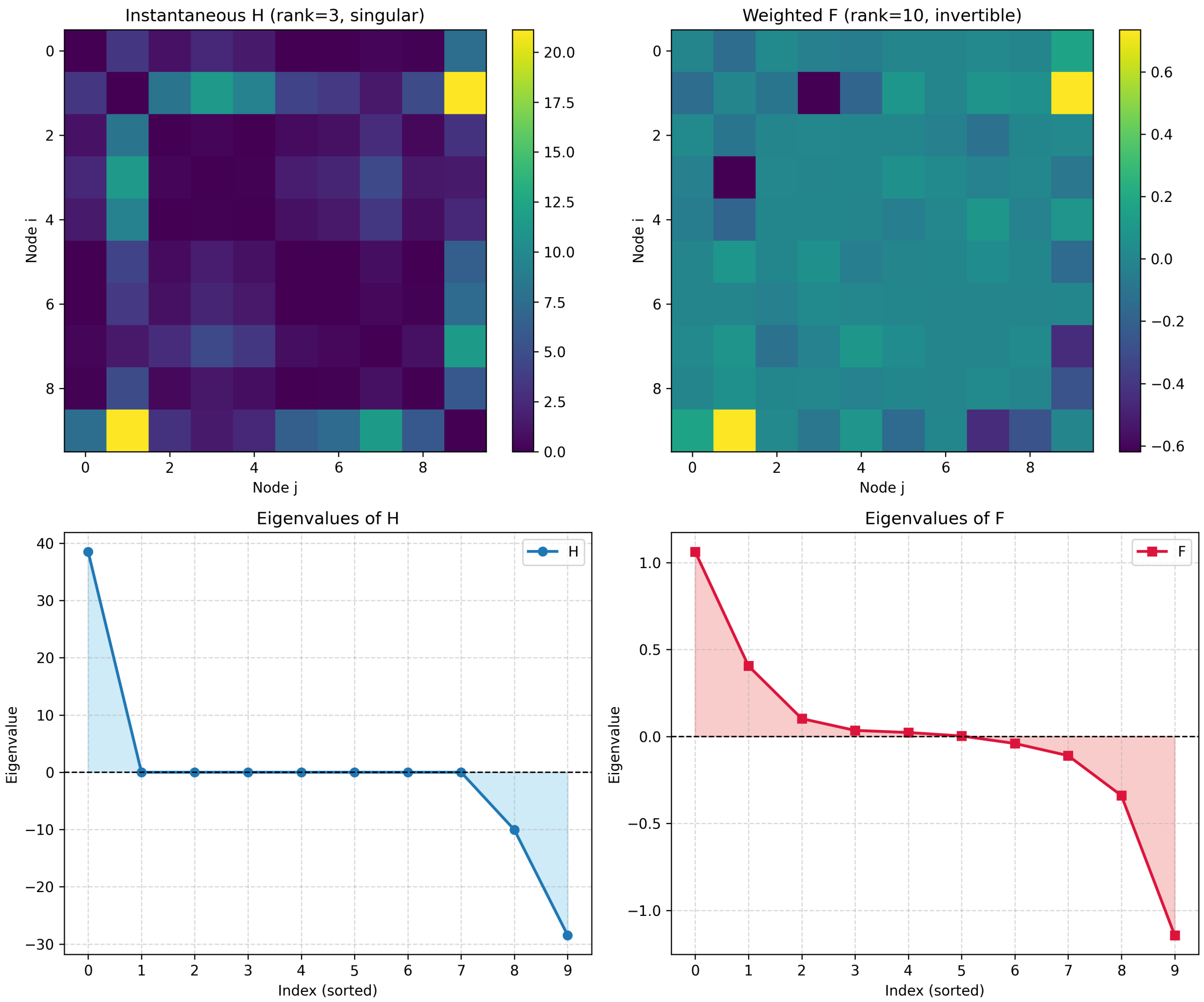

Figure 7: Top-left: the instantaneous squared-difference matrix H at a single time point, which is rank-deficient (rank $= 3$) and singular, showing large pairwise distances only for a few node pairs. Top-right: the Hadamard-weighted matrix $F = C \circ H$, where C is the long-term correlation; weighting lifts H to full rank (rank $= 10$) and makes F invertible. Bottom-left: the sorted eigenvalues of H, displaying exactly three nonzero modes and seven zeros, consistent with $\text{rank}(H) = 3$. Bottom-right: the sorted eigenvalues of F, all ten nonzero and of mixed sign, confirming that F is indefinite but invertible.

(2) $\det(J(t) \circ W)$ **is a nonzero polynomial.** By the Leibniz formula,

$$\det(\Omega(t)) = \sum_{\pi \in S_N} \text{sgn}(\pi) \prod_{i=1}^{N} \Omega(t)_{i,\pi(i)}$$

$$= \sum_{\substack{\pi \in S_N \\ \pi(i) \neq i \ \forall i}} \text{sgn}(\pi) \prod_{i=1}^{N} \big[J(t)_{i,\pi(i)} \, W_{i,\pi(i)}\big].$$

since $\Omega(t)_{ii} = 0$ kills any term with a fixed point. Thus

$$\det(J(t) \circ W) = \sum_{\substack{\pi \in S_N \\ \pi(i) \neq i}} \Big(\text{sgn}(\pi) \prod_{i=1}^{N} J(t)_{i,\pi(i)}\Big) \Big(\prod_{i=1}^{N} W_{i,\pi(i)}\Big),$$

a multivariate polynomial $P(\{W_{ij}\})$ in the off-diagonal W_{ij}.

To show $P \not\equiv 0$, pick the N-cycle $\pi_0\colon i \mapsto i+1 \pmod N$. Its monomial is

$$\prod_{i=1}^{N} W_{i,\pi_0(i)} = W_{1,2}\, W_{2,3} \cdots W_{N-1,N}\, W_{N,1},$$

and its coefficient is

$$\operatorname{sgn}(\pi_0) \prod_{i=1}^{N} J(t)_{i,\pi_0(i)} = \pm\,(x_1-x_2)^2\,(x_2-x_3)^2 \cdots (x_N-x_1)^2 \neq 0$$

because the x_i are distinct. Hence P has at least one nonzero coefficient and so is not the zero polynomial. Therefore it vanishes only on a proper hypersurface in $\mathcal{W}$, proving that for almost every full-support W, $\det(J(t) \circ W) \neq 0$ and $\operatorname{rank}(J(t) \circ W) = N$. □

A.9 Stability of GVNN Layer

Theorem 4 (GVNN Layer is Globally Lipschitz)**.** Let $W \in \mathbb{R}^{N\times N}$ be symmetric with nonnegative entries, and define

$$\alpha \;=\; \max_{1\le i\le N} \sum_{j=1}^{N} W_{ij}.$$

Let

$$X = [\,x(1)\;\ldots\;x(T)\,] \;\in\; \mathbb{R}^{N\times T},$$

and write

$$\mu_i = \frac{1}{T}\sum_{t=1}^{T} x_i(t),$$

$$M = \max_{\substack{1\le i\le N\\ 1\le t\le T}} \bigl|x_i(t)-\mu_i\bigr|,$$

$$B = \max_{\substack{1\le i\le N\\ 1\le t\le T}} \bigl|x_i(t)\bigr|.$$

Let scalar filters $a=(a_t)_{t=1}^{T}$ and $b=(b_t)_{t=1}^{T}$ satisfy

$$a^\star = \max_{1\le t\le T} |a_t|, \qquad b^\star = \max_{1\le t\le T} |b_t|.$$

For each t define two node functions:

$$J^{\mathrm{IC}}_{ij}(t) \;=\; \left|(x_i(t)-\mu_i)\,(x_j(t)-\mu_j)\right|, \qquad J^{\mathrm{SD}}_{ij}(t) \;=\; (x_i(t)-x_j(t))^2,$$

and form the Hadamard products

$$\Omega^{\mathrm{IC}}(t) \;=\; W \circ J^{\mathrm{IC}}(t), \qquad \Omega^{\mathrm{SD}}(t) \;=\; W \circ J^{\mathrm{SD}}(t).$$

Given any pointwise-1-Lipschitz nonlinearity $\sigma:\mathbb{R}\to\mathbb{R}$, define

$$y(t) \;=\; \sigma\bigl(a_t\,x(t) \;+\; b_t\,\Omega(t)\,x(t)\bigr), F(X) \;=\; \bigl[\,y(1)\;\ldots\;y(T)\bigr] \;\in\; \mathbb{R}^{N\times T}.$$

Then for every pair $X, X' \in \mathbb{R}^{N\times T}$,

$$\|F(X)-F(X')\|_F \;\le\; \bigl(a^\star \;+\; \alpha\, b^\star\, M^2\bigr)\,\|X-X'\|_F \quad \text{(IC)},$$

$$\|F(X)-F(X')\|_F \;\le\; \bigl(a^\star \;+\; 4\,\alpha\, b^\star\, B^2\bigr)\,\|X-X'\|_F \quad \text{(SD)}.$$

Proof. Because W is symmetric with nonnegative entries, Gershgorin's circle theorem guarantees that every eigenvalue λ of W lies in the interval $[0, \alpha]$. Consequently, the spectral (operator) norm of W satisfies

$$\|W\|_{\mathrm{op}} \;\le\; \alpha.$$

Fix an arbitrary time index $t \in \{1, \ldots, T\}$. We treat the IC and SD cases in parallel, noting only where the node function definition differs.

Define the diagonal matrix

$$D(t) \;=\; \operatorname{diag}\bigl(|x_i(t) - \mu_i|\bigr)_{i=1}^{N}.$$

By definition of M,

$$\|D(t)\|_{\mathrm{op}} = \max_{1 \le i \le N} \bigl|x_i(t) - \mu_i\bigr| \;\le\; M.$$

Since the Hadamard product with $G^{\mathrm{IC}}(t)$ coincides with the congruence

$$\Omega^{\mathrm{IC}}(t) = D(t)\, W\, D(t),$$

submultiplicativity of the operator norm yields

$$\begin{aligned}\|\Omega^{\mathrm{IC}}(t)\|_{\mathrm{op}} &\le \|D(t)\|_{\mathrm{op}} \cdot \|W\|_{\mathrm{op}} \cdot \|D(t)\|_{\mathrm{op}} \\ &\le M \cdot \alpha \cdot M = \alpha\, M^2.\end{aligned}$$

Here each entry of the instantaneous matrix is $(x_i(t) - x_j(t))^2$. We bound this directly in terms of the maximum node value B:

$$\begin{aligned}(x_i(t) - x_j(t))^2 &= \bigl|x_i(t) - x_j(t)\bigr|^2 \\ &\le \bigl(|x_i(t)| + |x_j(t)|\bigr)^2 \\ &\le (B + B)^2 = 4\, B^2.\end{aligned}$$

Therefore, for every i, j,

$$|\Omega^{\mathrm{SD}}_{ij}(t)| = W_{ij} \cdot (x_i(t) - x_j(t))^2 \;\le\; 4\, B^2\, W_{ij}.$$

Summing over j shows that each row sum of $\bigl|\Omega^{\mathrm{SD}}(t)\bigr|$ is at most $\sum_j 4\, B^2\, W_{ij} = 4\, B^2 \sum_j W_{ij} \le 4\,\alpha\, B^2$. Since $\Omega^{\mathrm{SD}}(t)$ remains symmetric with nonnegative entries, its operator norm is upper bounded by its maximum row sum, giving

$$\|\Omega^{\mathrm{SD}}(t)\|_{\mathrm{op}} = \rho\bigl(\Omega^{\mathrm{SD}}(t)\bigr) \;\le\; \max_{1 \le i \le N} \sum_{j=1}^{N} \Omega^{\mathrm{SD}}_{ij}(t) \;\le\; 4\,\alpha\, B^2.$$

In either case define the map $g_t : \mathbb{R}^N \to \mathbb{R}^N$ by

$$g_t(z) \;=\; a_t\, z \;+\; b_t\, \Omega(t)\, z.$$

For any two vectors $u, v \in \mathbb{R}^N$, we have

$$g_t(u) - g_t(v) \;=\; \bigl(a_t\, I + b_t\, \Omega(t)\bigr)\,(u - v).$$

Applying the triangle inequality together with the operator-norm bound on $\Omega(t)$ yields

$$\bigl\|g_t(u) - g_t(v)\bigr\|_2 \;\le\; |a_t|\; \|u - v\|_2 \;+\; |b_t|\; \|\Omega(t)\|_{\mathrm{op}}\; \|u - v\|_2.$$

Since $|a_t| \le a^\star$ and $|b_t| \le b^\star$, it follows that

$$\bigl\|g_t(u) - g_t(v)\bigr\|_2 \;\le\; \bigl(a^\star + b^\star\, \|\Omega(t)\|_{\mathrm{op}}\bigr)\; \|u - v\|_2.$$

Because σ is pointwise 1-Lipschitz, for each t and each pair of signals $x(t), x'(t)$,

$$\begin{aligned}\|y(t) - y'(t)\|_2 &= \bigl\|\sigma\bigl(g_t(x(t))\bigr) - \sigma\bigl(g_t(x'(t))\bigr)\bigr\|_2 \\ &\le \bigl\|g_t(x(t)) - g_t(x'(t))\bigr\|_2 \\ &\le L\, \|x(t) - x'(t)\|_2,\end{aligned}$$

where

$$L = \begin{cases} a^\star + \alpha\, b^\star\, M^2, & \text{IC}, \\ a^\star + 4\,\alpha\, b^\star\, B^2, & \text{LDE}. \end{cases}$$

Finally, summing these squared-norm inequalities over $t = 1, \dots, T$ and taking the square root gives:

$$\begin{aligned} \|F(X) - F(X')\|_F &= \left(\sum_{t=1}^{T} \|y(t) - y'(t)\|_2^2\right)^{1/2} \\ &\le L \left(\sum_{t=1}^{T} \|x(t) - x'(t)\|_2^2\right)^{1/2} \\ &= L\, \|X - X'\|_F. \end{aligned}$$

This completes the proof.

□

A.10 Extended Theorems and Proofs

This section will include more theorems and propositions for completeness

Theorem 5 (Parseval identity for the GVFT)**.** For every time index t and every signal vector $x(t)$, one has

$$\sum_{i=1}^{N} \left|\widehat{x}_i(t)\right|^2 = \sum_{i=1}^{N} \left|x_i(t)\right|^2.$$

Equivalently,

$$\|\widehat{x}(t)\|_2 = \|x(t)\|_2.$$

Proof. Due to Corollary 1, as long as the stable support is symmetric, the eigendecomposition of a connectivity profile results in U_t being orthonormal, i.e., $U_t^\top U_t = I_N$. Applying this to $\widehat{x}(t) = U_t^\top x(t)$, we get:

$$\begin{aligned} \|\widehat{x}(t)\|_2^2 &= \widehat{x}(t)^\top \widehat{x}(t) \\ &= (U_t^\top x(t))^\top (U_t^\top x(t)) \\ &= x(t)^\top U_t U_t^\top x(t) \\ &= x(t)^\top x(t) \\ &= \|x(t)\|_2^2. \end{aligned}$$

□

Remark 6. Because the GVFT basis U_t depends on the instantaneous, *signal-derived* slice $\Omega(t)$, Parseval's identity above holds *separately* for each time step t; summing over t shows energy conservation for the entire spatio-temporal matrix $X = [x(1) \dots x(T)]$:

$$\sum_{t=1}^{T} \|\widehat{x}(t)\|_2^2 = \sum_{t=1}^{T} \|x(t)\|_2^2.$$

The next theorem develops bounds on the eigenvalues of the instantaneous correlation node function profile againt a PSD stable support in terms of the eigenvalues of the PSD stable support.

Theorem 7 (IC Spectral bounds under amplitude-scaling)**.** Let

$$W \;\in\; \mathbb{S}_{++}^N \quad \text{have spectrum} \quad \lambda_{\min}(W) \le \cdots \le \lambda_{\max}(W),$$

and at time t let the centred sample $\tilde{x}_t \in \mathbb{R}^N$ satisfy $\tilde{x}_i(t) \neq 0$ for all i. Define

$$D_t = \operatorname{diag}\bigl(|\tilde{x}_t|\bigr), \quad \rho_t = D_t\, W\, D_t, \quad m_t = \min_i |\tilde{x}_i(t)|, \quad M_t = \max_i |\tilde{x}_i(t)|.$$

If $\delta_{1,t} \le \cdots \le \delta_{N,t}$ are the eigenvalues of ρ_t, then for each $i = 1, \dots, N$,

$$m_t^2\, \lambda_{\min}(W) \;\le\; \delta_{i,t} \;\le\; M_t^2\, \lambda_{\max}(W).$$

Proof. Recall the Rayleigh quotient of a symmetric matrix A and nonzero w is

$$\mathcal{R}(A; w) := \frac{w^\top A\, w}{w^\top w}.$$

By the Rayleigh–Ritz theorem (a special case of the Courant–Fischer min–max theorem),

$$\lambda_{\min}(A) \;\le\; \mathcal{R}(A; w) \;\le\; \lambda_{\max}(A), \quad \forall\, w \neq 0,$$

and the eigenvalues of A coincide with the extremal values of $\mathcal{R}(A; w)$ over appropriate subspaces.

For any unit vector $v \in \mathbb{R}^N$ ($\|v\| = 1$), consider

$$v^\top \rho_t\, v = v^\top (D_t\, W\, D_t)\, v = (D_t\, v)^\top W\, (D_t\, v).$$

We will bound $(D_t v)^\top W\, (D_t v)$ using $\mathcal{R}(W;\, \cdot)$.

Define

$$u \;=\; \frac{D_t\, v}{\|D_t\, v\|}, \quad u \neq 0, \quad \|u\| = 1.$$

Then

$$(D_t\, v)^\top W\, (D_t\, v) = \|D_t\, v\|^2\; \underbrace{\frac{(D_t v)^\top W\, (D_t v)}{\|D_t v\|^2}}_{=\mathcal{R}(W;u)}.$$

By the Rayleigh–Ritz result of Step 1,

$$\lambda_{\min}(W) \;\le\; \mathcal{R}(W; u) \;\le\; \lambda_{\max}(W),$$

so

$$(D_t v)^\top W\, (D_t v) \;\in\; \big[\lambda_{\min}(W)\, \|D_t v\|^2,\; \lambda_{\max}(W)\, \|D_t v\|^2\big].$$

Since v has $\|v\| = 1$ and $D_t = \operatorname{diag}(d_{1,t}, \ldots, d_{N,t})$ with $d_{i,t} = |\tilde{x}_i(t)| \in [m_t, M_t]$, we have

$$\|D_t v\|^2 = \sum_{i=1}^N d_{i,t}^2\, v_i^2 \;\in\; [\, m_t^2,\; M_t^2\,].$$

Therefore for every unit v,

$$v^\top \rho_t\, v = (D_t v)^\top W\, (D_t v) \;\in\; \big[m_t^2\, \lambda_{\min}(W),\; M_t^2\, \lambda_{\max}(W)\big].$$

Finally, by the Courant–Fischer characterization of eigenvalues, the ith largest eigenvalue $\delta_{i,t}$ of ρ_t is the extremal Rayleigh quotient over an i-dimensional subspace. Since *all* Rayleigh quotients lie in $\big[m_t^2 \lambda_{\min}(W),\, M_t^2 \lambda_{\max}(W)\big]$, each $\delta_{i,t}$ must also satisfy

$$m_t^2\, \lambda_{\min}(W) \;\le\; \delta_{i,t} \;\le\; M_t^2\, \lambda_{\max}(W), \quad i = 1, \ldots, N.$$

This completes the proof. □

Theorem 8 (IC Condition-number bound under amplitude-scaling)**.** Under the hypotheses of Theorem 1, let

$$d_i = |\tilde{x}_i^{(m)}(t)|, \quad d_{\min} = \min_{1 \le i \le N} d_i, \quad d_{\max} = \max_{1 \le i \le N} d_i,$$

and recall $W \in \mathbb{S}_{++}^N$ has spectrum $\lambda_{\min}(W) \le \cdots \le \lambda_{\max}(W)$. Then the instantaneous filtered matrix $\rho_t = D_t\, W\, D_t$ is SPD and its condition number satisfies

$$\kappa\big(\rho_t\big) \;=\; \frac{\lambda_{\max}\big(\rho_t\big)}{\lambda_{\min}\big(\rho_t\big)} \;\le\; \frac{d_{\max}^2}{d_{\min}^2} \;\cdot\; \frac{\lambda_{\max}(W)}{\lambda_{\min}(W)}.$$

Proof. From Theorem 1, $\rho_t^{(m)}$ is congruent to the SPD matrix W, so it remains SPD, hence all eigenvalues are strictly positive and the condition number is well-defined.

From the Rayleigh–Ritz characterization, for any unit vector v,

$$\begin{aligned} v^\top \rho_t v &= (D_t v)^\top W \, (D_t v) \\ &\in \big[\lambda_{\min}(W) \, \|D_t v\|^2, \ \lambda_{\max}(W) \, \|D_t v\|^2\big]. \end{aligned}$$

Since $d_{\min} \le d_i \le d_{\max}$ for all i, and $\|v\| = 1$, one checks

$$d_{\min}^2 \ \le \ \|D_t v\|^2 \ \le \ d_{\max}^2.$$

Hence every eigenvalue δ of $\rho_t^{(m)}$ satisfies

$$d_{\min}^2 \, \lambda_{\min}(W) \ \le \ \delta \ \le \ d_{\max}^2 \, \lambda_{\max}(W).$$

Writing $\delta_{\min} = \lambda_{\min}(\rho_t)$ and $\delta_{\max} = \lambda_{\max}(\rho_t)$, the above yields

$$\delta_{\min} \ \ge \ d_{\min}^2 \, \lambda_{\min}(W), \quad \delta_{\max} \ \le \ d_{\max}^2 \, \lambda_{\max}(W).$$

Therefore

$$\kappa\big(\rho_t\big) = \frac{\delta_{\max}}{\delta_{\min}} \ \le \ \frac{d_{\max}^2 \, \lambda_{\max}(W)}{d_{\min}^2 \, \lambda_{\min}(W)} \ = \ \frac{d_{\max}^2}{d_{\min}^2} \ \cdot \ \frac{\lambda_{\max}(W)}{\lambda_{\min}(W)},$$

which completes the proof. □

Theorem 9 (Gershgorin bounds on ρ_t)**.** Let $\rho_t \in \mathbb{S}_{++}^N$ as above, and define

$$a_{ii} = \rho_{ii} = W_{ii} \, d_{i,t}^2, \quad R_i = \sum_{j \ne i} |W_{ij}| \, d_i \, d_j.$$

Then every eigenvalue δ_i of ρ_t satisfies

$$\delta_i \ \in \ \bigcup_{i=1}^N D\big(a_{ii}, \, R_i\big) = \bigcup_{i=1}^N \big\{z : |z - W_{ii} \, d_i^2| \le d_i \sum_{j \ne i} |W_{ij}| \, d_j\big\}.$$

In particular, since W is SPD and its diagonal entries $W_{ii} > 0$, each disc lies strictly in the right-half plane and hence ρ_t has all positive eigenvalues.

Moreover, letting

$$d_{\min} = \min_i d_i, \quad d_{\max} = \max_i d_i, \quad r_{\max} = \max_i \sum_{j \ne i} |W_{ij}|,$$

we obtain the simplified bound

$$\delta_i \ \in \ \big[\, d_{\min}^2 \ \min_i W_{ii} - d_{\max}^2 \, r_{\max}, \ d_{\max}^2 \ \max_i W_{ii} + d_{\max}^2 \, r_{\max}\big].$$

Proof. By Gershgorin's circle theorem, each eigenvalue δ of $\rho = \rho_t$ lies in at least one disc

$$\{z : |z - \rho_{ii}| \le \sum_{j \ne i} |\rho_{ij}|\}, \quad \rho_{ii} = W_{ii} d_i^2, \quad \rho_{ij} = W_{ij} d_i d_j.$$

Thus

$$|\, \delta_i - W_{ii} d_i^2| \ \le \ d_i \sum_{j \ne i} |W_{ij}| \, d_j.$$

Since W is SPD, $W_{ii} = \sum_k \lambda_k u_{k,i}^2 > 0$ and each $d_i > 0$. Hence the real parts of all discs lie strictly to the right of zero, proving $\delta_i > 0$.

For the coarse bound, note

$$W_{ii} \ge \min_i W_{ii}, \quad \sum_{j \ne i} |W_{ij}| \le r_{\max}, \quad d_{\min} \le d_{i,t} \le d_{\max},$$

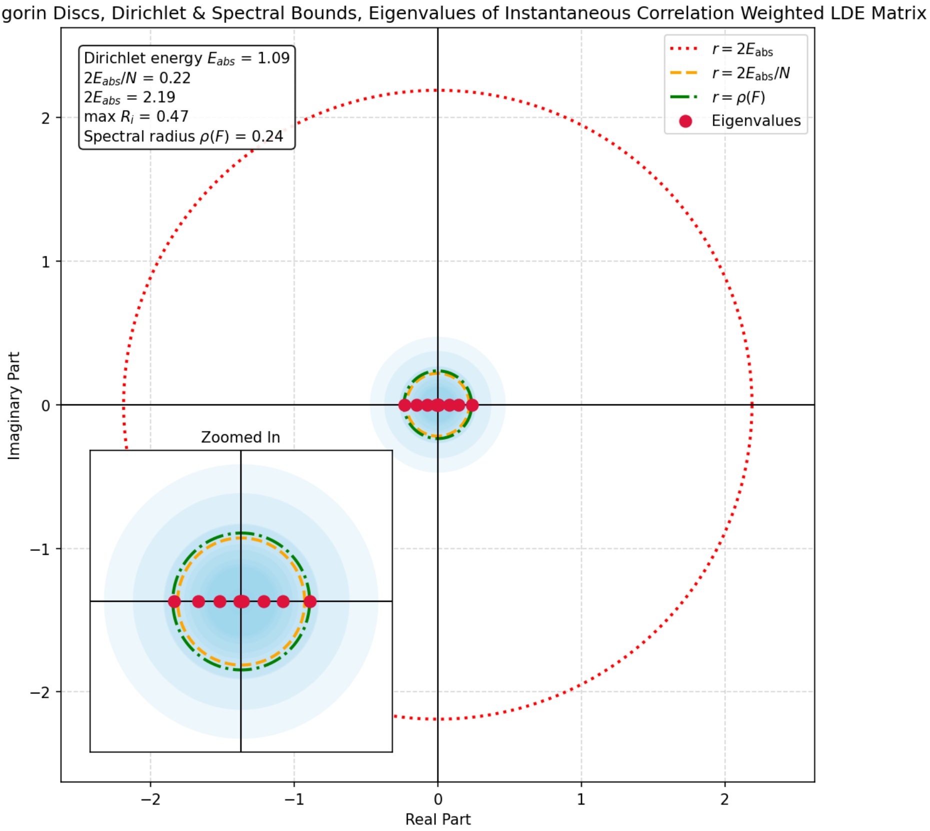

Figure 8: Main panel: Light-blue shaded circles show the Gershgorin discs of the Hadamard-weighted matrix $F = C \circ H$, each centered at the origin with radius $R_i = \sum_{j\neq i} |C_{ij}|(x_i - x_j)^2$. Red dotted circle marks the upper Dirichlet-energy bound $r = 2E_{\text{abs}}$, orange dashed circle marks the average-energy bound $r = 2E_{\text{abs}}/N$, and green dash–dot circle marks the spectral radius $r = \rho(F)$. Red crosses are the eigenvalues of F, all lying within the union of the Gershgorin discs. We can see how the spectral radius is upper bounded by the Dirichlet Energy.
Zoomed In: A close-up around the origin shows the small Gershgorin discs, the tight Dirichlet lower bound $2E_{\text{abs}}/N$, and the spectral-radius circle relative to the cluster of eigenvalues, we clearly see how $\max_i R_i$ strictly exceeds $2E_{\text{abs}}/N$ yet remains below $2E_{\text{abs}}$.

So every disc collapses to the real interval (as all eigenvalues are real):

$$\begin{aligned} &D\big(W_{ii}d_i^2,\, d_i \textstyle\sum_{j\neq i} |W_{ij}|\, d_j\big) \\ &\qquad \subset \big[d_{\min}^2 \min_i W_{ii} - d_{\max}^2 r_{\max},\; d_{\max}^2 \max_i W_{ii} + d_{\max}^2 r_{\max}\big]. \end{aligned}$$

Therefore all eigenvalues δ_i lie in the stated interval. □

Proposition 1. *Let $x_1(t), \ldots, x_N(t)$ be distinct real numbers,*

$$J_{ij}(t) = (x_i(t) - x_j(t))^2, \quad J_{ii}(t) = 0,$$

and let $W \in \mathbb{R}^{N\times N}$ be any symmetric matrix with $W_{ij} \neq 0$ for all $i \neq j$. Define the Hadamard product

$$\Omega(t) \;=\; J(t) \circ W, \qquad \Omega(t)_{ij} = J(t)_{ij}\, W_{ij}.$$

Then $\Omega(t)$ is symmetric and invertible, yet $\operatorname{tr}(\Omega(t)) = 0$, so $\Omega(t)$ cannot be positive (semi-)definite.

Proof. First, symmetry of $\Omega(t)$ follows immediately from symmetry of $J(t)$ and W, since

$$\Omega(t)_{ij} = J_{ij}(t)\, W_{ij} = J_{ji}(t)\, W_{ji} = \Omega(t)_{ji}.$$

Invertibility is guaranteed by the Hadamard rank-lifting argument: because W has full support, $\det(D \circ C) \neq 0$ for generic such W, hence $\operatorname{rank}(\Omega(t)) = N$.

Next, compute the trace:

$$\operatorname{tr}(\Omega(t)) = \sum_{i=1}^{N} \Omega(t)_{ii} = \sum_{i=1}^{N} J(t)_{ii}\, W_{ii} = \sum_{i=1}^{N} 0 \cdot W_{ii} = 0.$$

Finally, if M were positive semi definite then all its eigenvalues $\{\lambda_k\}$ would satisfy $\lambda_k \geq 0$. But their sum is

$$\sum_{k=1}^{N} \lambda_k = \operatorname{tr}(M) = 0,$$

forcing each $\lambda_k = 0$, contradicting invertibility. Hence M has both positive and negative eigenvalues and is indefinite. □

Theorem 10 (Spectral bounds for LDE weighting)**.** Let $C \in \mathbb{R}^{N\times N}$ be a real symmetric, full-rank matrix with eigenvalues

$$\lambda_{\min}(C) \;\leq\; \cdots \;\leq\; \lambda_{\max}(C).$$

At time t, let $x(t) \in \mathbb{R}^N$ and define the instantaneous squared-difference matrix

$$J_{ij}(t) = (x_i(t) - x_j(t))^2, \quad J_{ii}(t) = 0,$$

and form the Hadamard-weighted matrix

$$\Omega(t) = W \circ J(t), \quad \Omega_{ij}(t) = W_{ij}\, J_{ij}(t).$$

Set

$$m_t = \min_{i\neq j} |x_i(t) - x_j(t)|, \qquad M_t = \max_{i\neq j} |x_i(t) - x_j(t)|,$$

and let $\delta_{1,t} \leq \cdots \leq \delta_{N,t}$ be the eigenvalues of $\Omega(t)$. Then for each $i = 1, \ldots, N$,

$$m_t^2\, \lambda_{\min}(W) \;\leq\; \delta_{i,t} \;\leq\; M_t^2\, \lambda_{\max}(W).$$

Proof. Let $v \in \mathbb{R}^N$ be any unit vector, $\|v\| = 1$. The Rayleigh quotient of $\Omega(t)$ at v is

$$v^\top \Omega(t)\, v = \sum_{i,j} W_{ij}\, (x_i(t) - x_j(t))^2\, v_i\, v_j.$$

Since for all $i \neq j$ we have $m_t^2 \leq (x_i(t) - x_j(t))^2 \leq M_t^2$, it follows that

$$m_t^2 \sum_{i,j} W_{ij} v_i v_j \;\leq\; v^\top \Omega(t)\, v \;\leq\; M_t^2 \sum_{i,j} W_{ij} v_i v_j.$$

But $\sum_{i,j} W_{ij} v_i v_j = v^\top W\, v$, and by the Rayleigh–Ritz theorem

$$\lambda_{\min}(W) \leq v^\top W\, v \leq \lambda_{\max}(W).$$

Combining these inequalities gives

$$m_t^2\, \lambda_{\min}(W) \;\leq\; v^\top \Omega\, v \;\leq\; M_t^2\, \lambda_{\max}(W).$$

Finally, the Courant–Fischer characterization implies that each eigenvalue $\delta_{i,t}$ of $\Omega(t)$ lies within the range of $v^\top \Omega(t)\, v$ over unit v. Therefore

$$m_t^2\, \lambda_{\min}(W) \;\leq\; \delta_{i,t} \;\leq\; M_t^2\, \lambda_{\max}(W), \quad i = 1, \ldots, N,$$

as claimed. □

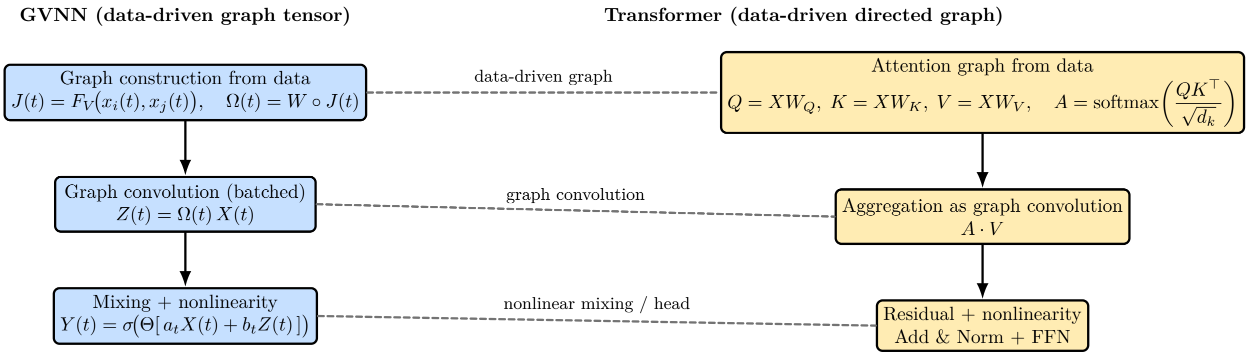
GVNN (data-driven graph tensor)
Transformer (data-driven directed graph)
Graph construction from data
$J(t) = F_V(x_i(t), x_j(t)), \quad \Omega(t) = W \circ J(t)$
data-driven graph
Attention graph from data
$Q = XW_Q,\ K = XW_K,\ V = XW_V, \quad A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)$
Graph convolution (batched)
$Z(t) = \Omega(t)\, X(t)$
graph convolution
Aggregation as graph convolution
$A \cdot V$
Mixing + nonlinearity
$Y(t) = \sigma\big(\Theta[\, a_t X(t) + b_t Z(t)\,]\big)$
nonlinear mixing / head
Residual + nonlinearity
Add & Norm + FFN

Figure 9: Both architectures construct a graph from the input and then convolve over it. GVNN forms a data-driven adjacency tensor $\Omega(t) = W \circ J(t)$ and performs $Z(t) = \Omega(t)X(t)$ before a learned mixing and nonlinearity $Y(t) = \sigma\big(\Theta[\, a_t X(t) + b_t Z(t)\,]\big)$. A Transformer builds a directed, data-driven attention graph $A = \mathrm{softmax}(QK^\top/\sqrt{d_k})$ and aggregates via $A \cdot V$, followed by residual connections and a feed-forward network.

A.11 Relation to LoRA and HiRA Adapters

Parameter–efficient fine-tuning (PEFT) adapts large models by training only a small number of parameters. **LoRA** (Hu et al., 2021) achieves this by expressing the update as a low–rank factorization, $\Delta W = AB$ with $\mathrm{rank}(\Delta W) \le r$, trading full expressiveness for efficiency. **HiRA** (Huang et al., 2025) increases expressiveness without sacrificing PEFT by applying a Hadamard (elementwise) product between a high–rank base and a low–rank factor:

$$\Delta W = W_0 \odot (AB), \quad \text{with} \quad \mathrm{rank}(\Delta W) \le \mathrm{rank}(W_0)\,\mathrm{rank}(AB).$$

This allows the update to attain a much higher effective rank while keeping trainable parameters comparable to LoRA.

GVNNs leverage the same algebraic idea. At each time step, an instantaneous (often low–rank) connectivity J_t is fused with a stable, typically high–rank support W via a Hadamard product, $\Omega_t = W \odot J_t$. This multiplicative fusion boosts the rank and stabilizes Ω_t, ensuring a more expressive operator even when J_t is rank–deficient.

In fact, the support W need not be fixed. In analogy to LoRA, one can parameterize W itself as

$$W = W_{\text{base}} + \Delta W, \qquad \Delta W = AB,$$

where W_{base} is an initialization (e.g., long–term correlation) and ΔW is a low–rank adapter. This formulation enables *efficient adaptation of the support* while avoiding the cost of learning a full $N \times N$ matrix. Alternatively, in a HiRA–style design, we may define

$$W = W_{\text{base}} \odot (AB),$$

so that the expressive capacity of the Hadamard product is preserved even when AB is low–rank.

This perspective shows that the Hadamard support in GVNNs can itself be learned using LoRA/HiRA adapters: low–rank updates capture task–specific variations, while the Hadamard structure ensures that these updates interact multiplicatively with instantaneous connectivities J_t. In practice, this allows GVNNs to scale to large graphs without incurring prohibitive parameter costs, while retaining the flexibility to adapt supports across datasets and tasks.

A.12 Transformers are Graph Variate Neural Networks (and vice-versa)

Recent work has suggested that the transformer model is in fact a graph neural network that has *'won the hardware lottery'*. This suggests that we can, in fact, go the other direction and build better Graph Neural Network architectures by leveraging ideas from the transformer model.

The following discussion will demonstrate that the transformer architecture is in fact not only a Graph Neural Network but in fact a *Graph Variate Neural Network*, i.e one that's core operation is an input dependent graph convolution. In fact, the transformer block can be reinterpreted as a GVNN with a static graph variable tensor (i.e the attention matrix replicated over all T) just with differences in normalization and Linear Weight projections.

A.12.1 Transformer Self-Attention as Directed Data-Driven Graph Convolution

Given token features $X \in \mathbb{R}^{T\times d}$, the Transformer computes *queries*, *keys*, and *values*

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V, \tag{15}$$

then forms a *row-stochastic, directed* attention matrix

$$A \;=\; \mathrm{softmax}\!\left(\frac{QK^\top}{\sqrt{d_k}}\right) \;\in\; \mathbb{R}^{T\times T}, \tag{16}$$

and aggregates values via

$$\mathrm{Attn}(X) \;=\; A\,V \;\in\; \mathbb{R}^{T\times d_v}. \tag{17}$$

Equations equation 16–equation 17 implement *graph convolution on a data-driven, directed graph* whose adjacency is A: each row of A defines outgoing edges from a token to all others with weights given by the softmax of similarities. Residual connections and a position-wise feed-forward network complete the encoder block.

Multi-head attention. For H heads with $A^{(h)}$ and $V^{(h)}$, the aggregation is $\mathrm{Concat}_h\big(A^{(h)}V^{(h)}\big)W_O$, a parallel sum of graph convolutions on H distinct data-driven graphs.

A.12.2 GVNN as Data-Driven Graph Convolution

GVNN constructs a *graph–variate tensor* via two ingredients:

1. A **node-wise similarity/interaction** functional $F_V : \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ producing

   $$J_{ij}(t) \;=\; F_V\big(x_i(t), x_j(t)\big) \quad\Rightarrow\quad J(t) \in \mathbb{R}^{N\times N}. \tag{18}$$

   Examples include the LDE and instantaneous correlation.
2. A **stable support** $W \in \mathbb{R}^{N\times N}$ (fixed or learned) that encodes long-term topology or sparsity. GVNN forms the pointwise (Hadamard) product

   $$\Omega(t) \;=\; W \circ J(t)\,, \tag{19}$$

   which gates/filters instantaneous interactions by the support.

Given $\Omega(t)$, GVNN performs a batched graph convolution of the current signal:

$$Z(t) \;=\; \Omega(t)\,X(t) \;\in\; \mathbb{R}^{N}. \tag{20}$$

A compact GVNN layer then mixes the original and aggregated signals followed by a nonlinearity:

$$Y(t) \;=\; \sigma\Big(\Theta\,[\,a_t\,X(t) \;+\; b_t\,Z(t)\,]\Big), \tag{21}$$

where $\Theta \in \mathbb{R}^{N\times N}$ is a learned linear map (or small MLP), and a_t, b_t are (optionally learned) scalar/broadcast coefficients. Stacking L layers yields $H^{(l)}(t)$ with $H^{(0)}(t) = X(t)$ and

$$\Omega^{(l)}(t) = W \circ J^{(l)}(t), \qquad J^{(l)}_{ij}(t) = F_V\big(h_i^{(l-1)}(t), h_j^{(l-1)}(t)\big).$$

Multi-node function convolution. Similar to multi-head attention one may aggregate convolutions with different node functions and stable supports.