NEAR-BALANCED SIGNED GRAPH LEARNING & FILTERING FOR DEPRESSION SCORE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies have shown that strongly anti-correlated signals often induce nearly balanced signed graphs containing only a small number of inconsistent negative cycles (frustrations). However, existing methods do not learn such statistics as near-balanced signed graphs nor exploit this graph structure for signal filtering. We formulate an optimization that jointly learns a near-balanced signed graph and filters signals by alternating between balanced and frustrated graph components. Specifically, we build a generalized expectation maximization (GEM) framework that alternates between graph learning (M-step) and signal estimation (E-step). The M-step constructs a near-balanced graph from learned features and then regulates frustration, while the E-step denoises a signal by minimizing a graph Laplacian regularizer, alternating between low-pass filtering on the balanced component and solving a linear system for the frustrated component. Unrolling the resulting iterations yields a lightweight transformer-like neural network with direct correspondence to the underlying optimization steps. Applied to fMRI-based depression score prediction, our constructed network matches state-of-the-art (SOTA) performance while reducing parameters to fewer than 5% of SOTA methods, outperforming both positive and full-balanced graph models.

1 INTRODUCTION

Natural statistics with strong pairwise anti-correlations are common, such as in brain signals (*e.g.*, EEG and fMRI), where local neural regions responsible for different functionalities are often activated at different times. Many of these statistics have *nearly balanced* signed graph structures—there exist few frustrated cycles of odd numbers of negative edges (Antal et al., 2005; Hu & Zheng, 2013; Rawlings & Friedkin, 2017; Gallo et al., 2024a). While this phenomenon has recently been analyzed theoretically (Marvel et al., 2009; Kirkley et al., 2019; Gallo et al., 2024b), given (typically limited) observed anti-correlated data, existing graph learning methods either ignore edge signs or enforce full balance, neglecting the small yet systematic frustrations present in real anti-correlated data. We show that a dominant eigen-mode of the data affinity matrix provides a sufficient condition for near-balance (Proposition 1). To close this gap, we learn near-balanced signed graphs and use their balanced and frustrated components for graph filtering and representation learning.

Building on advances in *graph signal processing* (GSP) (Ortega et al., 2018; Cheung et al., 2018) and algorithm unrolling (Monga et al., 2021), learned graph filtering algorithms have been unrolled into transformer-like neural net for image restoration (Thuc et al., 2024; Cai et al., 2026), traffic prediction (Qi et al., 2026), and EEG classification (Yao et al., 2026b). The crux is the learning of a graph structure appropriate for the target data and subsequent graph filtering: positive graph for images, directed graph for traffic, and full-balanced signed graph for EEG signals. However, a full-balanced signed graph is too restrictive to model frustrations in many anti-correlated statistics such as fMRI.[1]

In this paper, focusing on fMRI-based depression score prediction as a concrete application, we first formulate an optimization for the *pretext* denoising task that jointly learns a near-balanced signed graph and filters signals by alternating between balanced and frustrated graph components. Specifically, we formalize the alternating graph learning / filtering procedures in Thuc et al. (2024) as M-

[1]Section 3.1 shows empirically that EEG signals exhibit higher spectral concentration (lower frustration) than fMRI, so full-balanced graph models may suffice for EEG but are too restrictive for fMRI.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and E-steps in a *generalized expectation maximization* (GEM) framework (Dempster et al., 1977). In the M-step (graph learning), we initialize a signed graph from learned features, then regulate graph frustration, resulting in a near-balanced graph $\mathcal{G}^s$. In the E-step (graph signal filtering), we decompose the graph Laplacian matrix $\mathcal{L} = \mathcal{L}^B + \mathbf{F}$ into balanced and frustrated components, so that a spectral low-pass filter for the balanced part and a linear system for the frustrated part can be applied in an *alternating direction method of multipliers* (ADMM) inner-loop (Boyd et al., 2011). Unrolling the two steps repeatedly into neural layers with learnable parameters yields a lightweight and interpretable transformer-like neural net (Thuc et al., 2024).

Finally, we train denoisers as a pretext task to capture statistics of patient groups, and use their features to predict depression scores. Experimental results show that our network unrolled from our near-balanced graph filtering algorithm matches state-of-the-art (SOTA) in fMRI-based depression score prediction, while using fewer than 5% of the parameters of SOTA methods. Moreover, our near-balanced graph model outperforms previous positive (Thuc et al., 2024) and full-balanced graph models (Yao et al., 2026b).

In summary, our technical contributions are

1. We reformulate near-balance in anti-correlated data as a dominant eigen-mode of the affinity matrix, bound the resulting frustration (Proposition 1), and verify the condition empirically on EEG and fMRI data. Leveraging this, we learn a near-balanced signed graph $\mathcal{G}^s$ by initializing a signed graph from learned features and regulating graph frustration.
2. Given the learned near-balanced structure, we decompose a near-balanced graph Laplacian $\mathcal{L}$ into graph-balanced and frustrated components, $\mathcal{L}^B$ and $\mathbf{F}$, so that a low-pass filter and linear system update for the two respective parts can be optimized in an ADMM setup via different network parameters.
3. We formalize the alternating graph learning / graph filtering steps in Thuc et al. (2024); Qi et al. (2026); Yao et al. (2026b) in a GEM framework, enabling interpretation through established GEM theory and providing a principled optimization perspective.
4. By unrolling our near-balanced signed graph denoisers into lightweight transformer-like neural nets, we achieve SOTA depression score prediction while reducing parameter count by more than 95% compared to SOTA transformer-based implementations.

2 PRELIMINARIES

2.1 GRAPH SIGNAL PROCESSING DEFINITIONS

We define GSP definitions as follows. A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ contains a node set $\mathcal{V} = \{1, \ldots, N\}$ and an edge set $\mathcal{E}$, where $(i, j) \in \mathcal{E}$ means nodes $i, j \in \mathcal{V}$ are connected by an edge with weight $w_{i,j} \in \mathbb{R}$. Denote by $\mathbf{W} \in \mathbb{R}^{N \times N}$ an *adjacency matrix*, where $W_{i,j} = w_{i,j}$ if $(i, j) \in \mathcal{E}$, and $W_{i,j} = 0$ otherwise. Denote by $\mathbf{D} \in \mathbb{R}^{N \times N}$ a diagonal *degree matrix*, where $D_{i,i} = \sum_j W_{i,j}$. Denote by $\mathbf{L} \in \mathbb{R}^{N \times N}$ the *combinatorial graph Laplacian* $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$. If the graph is undirected, *i.e.*, $w_{i,j} = w_{j,i}, \forall i, j$, $\mathbf{W}$ and $\mathbf{L}$ are symmetric. If self-loops exist, *i.e.*, $w_{i,i} \neq 0, \exists i$, then the *generalized graph Laplacian* $\mathcal{L} \triangleq \mathbf{D} - \mathbf{W} + \mathrm{diag}(\mathbf{W})$ is used. For signed graph $\mathcal{G}^s$ with positive and negative edges, the *signed graph Laplacian* $\mathbf{L}^s \triangleq \mathbf{D}^s - \mathbf{W}$ is often used, where the diagonal *absolute degree matrix* $\mathbf{D}^s$ has entries $D^s_{i,i} = d^s_i = \sum_j |W_{i,j}|$ (Kunegis et al., 2010). We will use both conventional Laplacian $\mathbf{L}$ (or $\mathcal{L}$) and signed Laplacian $\mathbf{L}^s$ to characterize a signed graph $\mathcal{G}^s$ in the sequel.

$\mathbf{L}$ (or $\mathcal{L}$) is provably *positive semi-definite* (PSD) if all edge weights are positive, *i.e.*, $W_{i,j} \geq 0, \forall i, j$ (Cheung et al., 2018). For a signed graph with both positive and negative edges, $\mathbf{L}$ (or $\mathcal{L}$) may be indefinite. Given $\mathbf{L}$ is real and symmetric, it can be eigen-decomposed into $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ by the Spectral Theorem (Horn & Johnson, 2012), where $\mathbf{V} \in \mathbb{R}^{N \times N}$ is the eigen-matrix with orthonormal eigenvectors $\{\mathbf{v}_k\}_{k=1}^N$ of $\mathbf{L}$ as columns, and $\mathbf{\Lambda} = \mathrm{diag}(\{\lambda_k\}) \in \mathbb{R}^{N \times N}$ is a diagonal matrix with ordered eigenvalues $\{\lambda_k\}_{k=1}^N$ along its diagonal. Assuming the graph $\mathcal{G}$ is positive, it is common in GSP to interpret the k-th eigen-pair $(\lambda_k, \mathbf{v}_k)$ of PSD $\mathbf{L}$ as the k-th frequency and Fourier mode for $\mathcal{G}$. Given a *graph signal* $\mathbf{x} \in \mathbb{R}^N$, $\boldsymbol{\alpha} = \mathbf{V}^\top \mathbf{x}$ is the *Graph Fourier Transform* (GFT), where α_k is the k-th GFT coefficient of signal $\mathbf{x}$.

2.2 GRAPH BALANCE

Unlike a positive graph $\mathcal{G}^+$ that models pairwise *attraction* (correlation) between nodes, a signed graph $\mathcal{G}^s$ with negative edges as well can model pairwise *repulsion* (anti-correlation). By definition, a signed graph $\mathcal{G}^s$ is balanced (denoted by $\mathcal{G}^B$) if there is no cycle with an odd number of negative edges. Instead of examining an exponential number of cycles in a graph, one can consider *node polarities*. Specifically, by the *Cartwright-Harary's Theorem* (CHT) (Harary, 1953), a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is balanced if each node $i \in \mathcal{V}$ can be assigned a polarity $\beta_i \in \{1, -1\}$ such that

$$\text{sign}(w_{i,j}) = \beta_i \beta_j, \quad \forall (i,j) \in \mathcal{E}. \tag{1}$$

One can now verify if a sparse signed graph $\mathcal{G}^s$ with $\mathcal{O}(N)$ edges is balanced, given polarity assignment $\boldsymbol{\beta} \in \{1, -1\}^N$, in complexity $\mathcal{O}(N)$ by checking (1) for all edges $(i,j) \in \mathcal{E}$.

2.3 MAPPING BETWEEN BALANCED SIGNED GRAPH AND POSITIVE GRAPH

It has been recently shown (Yang et al., 2022) that there exists a one-to-one mapping between a balanced signed graph $\mathcal{G}^B$ and a corresponding positive graph $\mathcal{G}^+$. First, denote by $\mathcal{L}^B$ the generalized graph Laplacian of a balanced signed graph $\mathcal{G}^B(\mathcal{V}, \mathcal{E}^B)$, and $\boldsymbol{\beta}$ is the polarity assignment for nodes $\mathcal{V}$ in $\mathcal{G}^B$, such that (1) is satisfied for all edges $(i,j) \in \mathcal{E}^B$. Define a diagonal matrix $\mathbf{S} \triangleq \text{diag}(\boldsymbol{\beta})$. Then, the generalized graph Laplacian $\mathcal{L}^+$ of the corresponding positive graph $\mathcal{G}^+(\mathcal{V}, \mathcal{E}^+)$ is a *similarity transform* of $\mathcal{L}^B$ (and thus share the same eigenvalues):

$$\mathcal{L}^+ = \mathbf{S}\mathcal{L}^B\mathbf{S}^{-1} \tag{2}$$

where $\mathbf{S} = \mathbf{S}^{-1}$. Thus, $\mathcal{G}^B$ and $\mathcal{G}^+$ have the same graph spectrum. Moreover, the eigen-matrices (sets of eigenvectors) are related: $\mathbf{V}^+ = \mathbf{S}\mathbf{V}^B$. Given that there exist substantial filter designs for positive graphs in GSP (Ortega et al., 2018), the transformation in (2) implies that positive graph filters can be reused to process signals on balanced signed graphs, as done in Yao et al. (2026a).

3 ALGORITHM DEVELOPMENT

3.1 SPECTRAL THEORY ON GRAPH NEAR-BALANCE

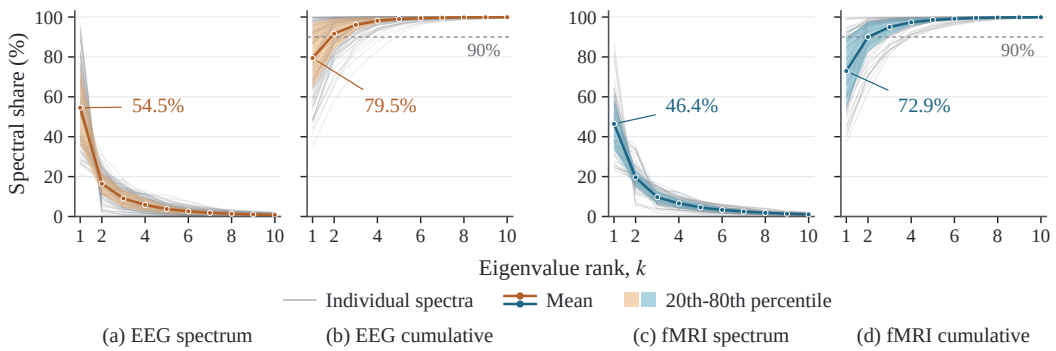
We start with the premise that many natural statistics with strong anti-correlations exhibit near-balanced graph structures, as empirically observed (Antal et al., 2005; Hu & Zheng, 2013; Rawlings & Friedkin, 2017; Gallo et al., 2024a). To explain this phenomenon, while previous works present noisy signed network arguments (Marvel et al., 2009; Kirkley et al., 2019; Gallo et al., 2024b), we instead establish the following sufficient condition: a constructed sparse graph with confidence filtering tends to be near-balanced, if an underlying affinity matrix $\mathbf{A}$ exhibits strong *spectral concentration*, *i.e.*, a dominant eigen-mode corresponding to λ_1 such that $\lambda_1 \gg \lambda_i, \forall i \neq 1$. $\mathbf{A}$ can be a covariance matrix, signed distance matrix, Gram matrix, etc, using which a sparse signed graph is constructed, as commonly done in the GSP literature (Ortega et al., 2018; Cheung et al., 2018).

To demonstrate that affinity matrix $\mathbf{A}$ with spectral concentration is common in practice, we examine the empirical covariance matrix $\mathbf{C}$ for EEG and fMRI signals, respectively. As shown in Fig. 1, we observe that the spectral energy of $\mathbf{C}$ is mostly concentrated in the first eigen-mode $\mathbf{v}_1$ corresponding to the first eigenvalue λ_1, where $\lambda_1 \geq \lambda_i, \forall i$. Notice also that EEG statistics have stronger spectral concentration (79.5%) than fMRI statistics (72.9%).

We next develop a theory to explain how spectral concentration leads to a sparse near-balanced signed graph. We first define *balanced template*: polarity assignments $\{\beta_i\}$ to graph nodes based on the first eigenvector $\mathbf{v}_1$ of $\mathbf{A}$, such that a graph containing edges with $\text{sign}(v_{1,i}v_{1,j})$ is balanced.

Definition 1 (Balanced template)**.** *Let* $\{(\lambda_k, \mathbf{v}_k)\}_{k=1}^N$ *be the eigen-pairs of* $\mathbf{A}$*, and let* $\beta_i \triangleq \text{sign}(v_{1,i}) \in \{\pm 1\}$ *(with* $\text{sign}(0) := 1$*). The* balanced template *of* $\mathbf{A}$ *is the signed graph on all node pairs with edge signs* $S^\star_{ij} \triangleq \beta_i\beta_j$*. It is balanced by CHT, since polarities* $\{\beta_i\}_{i=1}^N$ *satisfy* (1) *on every edge. Note that* $S^\star_{ij} = \text{sign}(v_{1,i}v_{1,j})$ *whenever* $v_{1,i}v_{1,j} \neq 0$*.*

We next define a *confidence-filtered graph* $\mathcal{G}_\tau$. Given a threshold $\tau > 0$, $\mathcal{G}_\tau$ has edge set $\mathcal{E}_\tau = \{(i,j) \,|\, i < j,\ |A_{i,j}| > \tau\}$ and edge signs $\sigma_{ij} = \text{sign}(A_{i,j})$. An edge $(i,j) \in \mathcal{E}_\tau$ is *frustrated* if



(a) EEG spectrum (b) EEG cumulative (c) fMRI spectrum (d) fMRI cumulative

Figure 1: Spectral concentration of EEG and fMRI covariance matrices: (a,c) normalized eigenvalues and (b,d) cumulative squared spectral energy. Gray curves show individual spectra, colored lines show means, and shaded bands indicate the 20th–80th percentiles.

and only if its sign is flipped relative to the template, *i.e.*, $\sigma_{i,j} = -S^{\star}_{i,j}$. Denote by $\mathcal{F}_\tau$ the set of frustrated edges and $f_\tau \triangleq |\mathcal{F}_\tau|/|\mathcal{E}_\tau|$. Thus, $\mathcal{G}_\tau$ is the balanced template restricted to $\mathcal{E}_\tau$ with the signs of the edges in $\mathcal{F}_\tau$ flipped. In particular, the frustration index of $\mathcal{G}_\tau$ is at most $|\mathcal{F}_\tau|$, and $\mathcal{G}_\tau$ is balanced if $\mathcal{F}_\tau = \emptyset$.

Proposition 1 (Near-balance from a dominant eigen-mode)**.** *Let* $\mathbf{A} \in \mathbb{R}^{N\times N}$ *be symmetric PSD with eigen-pairs* $(\lambda_k, \mathbf{v}_k)$*, orthonormal* $\mathbf{v}_k$*,* $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_N \geq 0$*, and eigen-gap* $\Delta \triangleq \lambda_1 - \lambda_2$*. Then for every* $\tau > 0$*:*

(i) (No flips for large τ*.) If* $\tau \geq (\lambda_1 - \Delta)/2$*, then* $\mathcal{F}_\tau = \emptyset$*, i.e.,* $\mathcal{G}_\tau$ *is balanced.*

(ii) (Few flips otherwise.) $|\mathcal{F}_\tau| \leq \dfrac{N(\lambda_1 - \Delta)^2}{8\tau^2}$*, i.e., the average number of frustrated edges per node is at most* $(\lambda_1 - \Delta)^2/(4\tau^2)$*.*

(iii) (Frustration rate.) Suppose moreover $\min_i |v_{1,i}| \geq c/\sqrt{N}$ *for some* $c \in (0, 1]$ *and* $\tau < \lambda_1 c^2/(2N)$*, and let* $\varepsilon \triangleq \dfrac{(\lambda_1 - \Delta)^2}{4(N-1)\tau^2} < 1$*. Then* $f_\tau \leq \dfrac{\varepsilon}{1-\varepsilon}$*.*

See Appendix A for a formal proof. Proposition 1 provides motivation to study near-balanced signed graph structures for natural statistics with spectral concentration, which we do rigorously next.

3.2 Generalized Expectation Maximization (GEM) Framework

As done in Yao et al. (2026b), to capture a target signal's prior distribution for a downstream application such as depression score prediction, we first design and train a denoiser for a pretext denoising task. Specifically, we design an alternating algorithm in an *generalized expectation maximization* (GEM) framework (Dempster et al., 1977), which unrolls into a transformer-like neural net[2]. In the *M-step*, we update parameters $\boldsymbol{\theta}$ that specify a near-balanced signed graph $\mathcal{G}^s$, represented by a PSD, real, and symmetric Laplacian matrix $\mathcal{L}_{\boldsymbol{\theta}}$, using latent signal $\mathbf{x}$. In the *E-step*, we compute latent signal $\mathbf{x}$ from noisy observation $\mathbf{y}$ and $\mathcal{L}_{\boldsymbol{\theta}}$ via a *maximum a posteriori* (MAP) formulation. Each pair of M- and E-steps will be unrolled into one neural layer for parameter tuning in the sequel.

3.3 M-step: Optimizing a Near-Balanced Graph

In a GEM framework, the *complete-data likelihood* is

$$p(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = p(\mathbf{y} \mid \mathbf{x})\, p(\mathbf{x}; \boldsymbol{\theta}) \tag{3}$$

[2]See Appendix C for details of unrolling a learned graph filtering algorithm into a transformer-like network.

where $p(\mathbf{y} \mid \mathbf{x})$ is the *observation likelihood*, and $p(\mathbf{x}; \boldsymbol{\theta})$ is the *prior distribution*. We define these terms respectively as iid Gaussian noise and *Gaussian Markov random field* (GMRF) models:

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp(-\frac{1}{\sigma^2}\|\mathbf{y} - \mathbf{x}\|_2^2) \tag{4}$$

$$p(\mathbf{x}; \boldsymbol{\theta}) \propto (\det{}^\dagger(\mathcal{L}_{\boldsymbol{\theta}}))^{\frac{1}{2}} \exp(-\frac{1}{2}\mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}} \mathbf{x}) \tag{5}$$

where $\mathcal{L}_{\boldsymbol{\theta}}$ is a PSD Laplacian matrix for a near-balanced signed graph $\mathcal{G}^s$ parametrized by $\boldsymbol{\theta}$ (to be detailed), and $\det{}^\dagger$ denotes the *pseudo-determinant*, *i.e.*, the product of the non-zero eigenvalues.

In the M-step, we maximize the complete-data likelihood:

$$\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} p(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} p(\mathbf{y} \mid \mathbf{x})\, p(\mathbf{x}; \boldsymbol{\theta}) \tag{6}$$

$$\overset{(a)}{=} \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} p(\mathbf{x}; \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} -\log p(\mathbf{x}; \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}} \mathbf{x} - \log\det{}^\dagger(\mathcal{L}_{\boldsymbol{\theta}}) \tag{7}$$

where (a) follows since $p(\mathbf{y} \mid \mathbf{x})$ in (4) does not depend on $\boldsymbol{\theta}$. $\boldsymbol{\Theta}$ defines a space of parameters $\boldsymbol{\theta}$ that defines valid Laplacian matrices for undirected signed graphs.

3.3.1 Initializing Signed Edge Weights

To compute Laplacian $\mathcal{L}_{\boldsymbol{\theta}}$ of a near-balanced signed undirected graph $\mathcal{G}^s$ in the M-step objective (7), we first initialize edge weights $\{w_{i,j}\}$ that encode *data geometry*—pairwise distances of samples $\{x_i\}$ in feature space—as follows. We compute *feature vector* $\mathbf{f}_i = F(\mathbf{e}_i) \in \mathbb{R}^K$ via a learned nonlinear function $F(\cdot)$ (to be detailed) for each node i, where $\mathbf{e}_i \in \mathbb{R}^M$ is an *embedding* for node i. Given $\mathbf{f}_i, \mathbf{f}_j$ for nodes i and j, we next compute non-negative *feature distance* $d_{i,j}$—Mahalanobis distance between samples i and j in the learned K-dimensional feature space for PSD *metric matrix* $\mathbf{M} \succeq 0$. Finally, signed edge weight $w_{i,j}$ is computed using the hyperbolic tangent function of $d_{i,j}$,

$$d_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M} (\mathbf{f}_i - \mathbf{f}_j), \qquad w_{i,j} = \tanh\left(\frac{d^\star - d_{i,j}}{\xi}\right), \tag{8}$$

where $d^\star > 0$ is the switching point beyond which $w_{i,j}$ becomes negative, and $\xi > 0$ is a shape parameter. Note that since $w_{i,j} \in [-1, 1]$ is a monotonic transformation of distance $d_{i,j}$, signed graph $\mathcal{G}^s$ preserves data geometry in its edge encoding. Signed edge weight assignment in (8) can be understood from a mass-spring system perspective (Knyazev, 2018): $d_{i,j} < d^\star$ implies mutual *attraction*, while $d_{i,j} > d^\star$ implies mutual *repulsion*. See Appendix B for a discussion.

3.3.2 Frustration-Minimizing Node Polarities

The initialized graph $\mathcal{G}^s$ may be too frustrated to be near-balanced. To regulate frustration, we perform the following. We first compute polarities $\beta_i \in \{1, -1\}$ to nodes i that minimize *frustration*—a weighted sum of edge weights $w_{i,j}$ that violate graph balance in (1), *i.e.*, $\text{sign}(w_{i,j}) \neq \beta_i \beta_j$:

$$\min_{\boldsymbol{\beta} \mid \beta_i \in \{1,-1\}} \sum_{i<j} |w_{i,j}| (\beta_i - \text{sign}(w_{i,j})\beta_j)^2. \tag{9}$$

Instead of solving (9), which is combinatorial[3], we solve the following relaxed version, whose solution is the first eigenvector $\mathbf{v}^*$ of $\mathbf{L}^s_{\boldsymbol{\theta}}$, the *signed graph Laplacian* matrix[4] for $\mathcal{G}^s$:

$$\mathbf{v}^* = \arg\min_{\mathbf{v} \mid \|\mathbf{v}\|=1} \mathbf{v}^\top \mathbf{L}^s_{\boldsymbol{\theta}} \mathbf{v} = \arg\min_{\mathbf{v} \mid \|\mathbf{v}\|=1} \sum_{i<j} |w_{i,j}| (v_i - \text{sign}(w_{i,j}) v_j)^2. \tag{10}$$

$\mathbf{v}^*$ can be computed using LOBPCG (Knyazev, 2001) in linear time, assuming $\mathbf{L}^s_{\boldsymbol{\theta}}$ is sparse. We extract $\mathbf{v}^*$'s entry signs for polarities: $\beta_i \leftarrow \text{sign}(v_i^*)$. Assigned polarities $\{\beta_i\}$ lead to identification of *frustrated edge set* $\mathcal{E}_F$: connected node pairs (i, j) where $\text{sign}(w_{i,j}) \neq \beta_i \beta_j$.

[3] See Appendix D for details of an alternative approach to compute polarities based on Gremban expansion.

[4] See Appendix E for a discussion on why the signed graph Laplacian $\mathbf{L}^s_{\boldsymbol{\theta}}$ is used for polarity assignment and frustration regulation as opposed to the combinatorial Laplacian $\mathbf{L}_{\boldsymbol{\theta}}$.

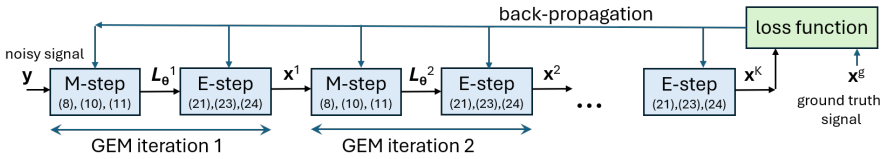


Figure 2: Overview of unrolled GEM network for a pretext signal denoising task.

3.3.3 Frustration Regulation

Proposition 1 motivates us to assume that the underlying graph $\mathcal{G}^s$ is near-balanced. Given this hyper-prior, to regulate frustration we reduce weight magnitude $m_{i,j}$ of each frustrated edge $w_{i,j} = s_{i,j}m_{i,j}$, where $s_{i,j} \in \{1, -1\}$ is the edge sign and $m_{i,j} \geq 0$ is the edge magnitude. Specifically, we optimize the following proxy of the M-step objective (7) that is a weighted sum of the prior and a regularization term that penalizes the magnitude of frustrated edge weights:

$$\min_{m_{i,j} \mid (i,j) \in \mathcal{E}_F} \mathbf{x}^\top \mathbf{L}_{\boldsymbol{\theta}}^s \mathbf{x} - \sum_{i \in \mathcal{V}} \log d_i^s + \upsilon \sum_{(i,j) \in \mathcal{E}_F} m_{i,j}, \quad \text{s.t.} \begin{cases} w_{i,j} = s_{i,j} m_{i,j} \\ s_{i,j} = \operatorname{sign}(w_{i,j}^o) \\ m_{i,j} = [0, |w_{i,j}^o|] \end{cases} \tag{11}$$

where $\upsilon \in \mathbb{R}_+$ is a positive tradeoff parameter, and $w_{i,j}^o$ is the initialized edge weight from Section 3.3.1. Following Kalofolias (2016), we replace $\log \det^\dagger(\mathbf{L}_{\boldsymbol{\theta}}^s)$ with the surrogate $\sum_{i \in \mathcal{V}} \log d_i^s$, where $d_i^s \geq 0$ by definition, for ease of computation. Note that the objective in (11) is convex, though the first two terms (denoted by $h_1(\cdot)$) are smooth while the last term (denoted by $h_2(\cdot)$) is non-smooth.

To minimize (11), we take a *proximal gradient descent* (PGD) approach (Parikh & Boyd, 2013). The derivative of the smooth part ($h_1(\cdot)$) w.r.t. $m_{i,j}$ is

$$\nabla_{m_{i,j}} h_1 \triangleq \frac{\partial h_1}{\partial m_{i,j}} = (x_i - s_{i,j} x_j)^2 - \left(\frac{1}{d_i^s} + \frac{1}{d_j^s} \right). \tag{12}$$

At each iteration t, each frustrated edge $(i,j) \in \mathcal{E}_F$ thus can make progress via $\tilde{m}_{i,j}^{t+1} = m_{i,j}^t - \eta \nabla_{m_{i,j}} h_1$, where $\eta > 0$ is a step size. Since $m_{i,j} \geq 0$, the proximal mapping for the non-smooth ℓ_1-norm term is the one-sided soft-thresholding operator:

$$\hat{m}_{i,j}^{t+1} = \max\left(\tilde{m}_{i,j}^{t+1} - \eta\upsilon, 0\right). \tag{13}$$

Finally, we project $\hat{m}_{i,j}^{t+1}$ back into the interval $[0, |w_{i,j}^o|]$ to obtain $m_{i,j}^{t+1}$, and set $w_{i,j}^{t+1} = s_{i,j} m_{i,j}^{t+1}$. We perform a finite T number of iterations to ensure progress in the E-step of a GEM framework.

3.3.4 Ensuring PSDness

Finally, given the resulting signed graph $\mathcal{G}^s$, we ensure that the generalized graph Laplacian $\mathcal{L}_{\boldsymbol{\theta}}$ is PSD so that the E-step quadratic regularizer $\mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}} \mathbf{x}$ is convex and bounded below. Specifically, we perform PSD-enforcing *diagonal loading* $\mathcal{L}_{\boldsymbol{\theta}} = \mathbf{L}_{\boldsymbol{\theta}} + \delta \mathbf{I}$, where $\delta = \max(-\lambda_{\min}(\mathbf{L}_{\boldsymbol{\theta}}), 0)$. The smallest eigenvalue $\lambda_{\min}(\mathbf{L}_{\boldsymbol{\theta}})$ can be computed efficiently using LOBPCG. Alternatively, a lower bound $\lambda_{\min}^-(\mathbf{L}_{\boldsymbol{\theta}})$ can be computed using *Gershgorin Circle Theorem* (GCT) (Varga, 2004).

3.4 E-step: Filtering on a Near-Balanced Graph

In the E-step, we maximize the complete-data likelihood via latent signal $\mathbf{x}$ for fixed Laplacian $\mathcal{L}_{\boldsymbol{\theta}}$:

$$\mathbf{x}^{t+1} = \arg\max_{\mathbf{x}} p(\mathbf{y}; \mathbf{x}; \boldsymbol{\theta}) = \arg\max_{x} p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}; \boldsymbol{\theta}) \tag{14}$$

$$= \arg\min - \log p(\mathbf{y} \mid \mathbf{x}) - \log p(\mathbf{x}; \boldsymbol{\theta}) = \arg\min \|\mathbf{y} - \mathbf{x}\|_2^2 + \mu\, \mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}} \mathbf{x} \tag{15}$$

where $\mu = \frac{\sigma^2}{2}$ is a weight parameter trading off the observation likelihood and the prior. (15) is a denoising problem for denoised signal $\mathbf{x} \in \mathbb{R}^N$ given noisy observation $\mathbf{y} \in \mathbb{R}^N$. (15) is an unconstrained quadratic program with a convex objective, and thus one approach is to simply solve the corresponding linear system $(\mathbf{I} + \mu\, \mathcal{L}_{\boldsymbol{\theta}})\mathbf{x}^* = \mathbf{y}$ for solution $\mathbf{x}^*$.

Instead, given that the balance component $\mathcal{L}_{\boldsymbol{\theta}}^{B}$ in $\mathcal{L}_{\boldsymbol{\theta}}$ has a well-defined notion graph frequencies (Dinesh et al., 2025), and we can efficiently compute $\mathbf{x}^*$ via graph spectral filter approximation (Susnjara et al., 2015). Hence, we divide near-balanced Laplacian $\mathcal{L}_{\boldsymbol{\theta}}$ into a balanced signed graph $\mathcal{L}_{\boldsymbol{\theta}}^{B}$ and a frustration component[5] $\mathbf{F}_{\boldsymbol{\theta}}$ (a Laplacian matrix containing only frustrated edges): $\mathcal{L}_{\boldsymbol{\theta}} = \mathcal{L}_{\boldsymbol{\theta}}^{B} + \mathbf{F}_{\boldsymbol{\theta}}$. The objective thus becomes

$$\min_{\mathbf{x}} \ \underbrace{\|\mathbf{y}-\mathbf{x}\|_2^2 + \mu\,\mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}}^{B}\mathbf{x}}_{f(\mathbf{x})} + \underbrace{\mu\,\mathbf{x}^\top \mathbf{F}_{\boldsymbol{\theta}}\mathbf{x}}_{g(\mathbf{x})} = f(\mathbf{x}) + g(\mathbf{x}). \tag{16}$$

We minimize (16) via an *alternating direction method of multipliers* (ADMM) framework (Boyd et al., 2011). First, we introduce an auxiliary variable $\mathbf{z} \in \mathbb{R}^N$ and rewrite (16) as

$$\min_{\mathbf{x},\mathbf{z}} \ f(\mathbf{x}) + g(\mathbf{z}), \quad \text{s.t. } \mathbf{x} = \mathbf{z}. \tag{17}$$

Then, using augmented Lagrange multipliers, we rewrite (17) as

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\tau}^\top(\mathbf{x}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}-\mathbf{z}\|_2^2 \tag{18}$$

where $\boldsymbol{\tau} \in \mathbb{R}^N$ is the Lagrange multiplier vector, and $\rho \in \mathbb{R}_+$ is a positive ADMM parameter. To solve (18), we update $\mathbf{x}, \mathbf{z}, \boldsymbol{\tau}$ alternately.

3.4.1 Solving for Primary Variable $\mathbf{x}$

When $\mathbf{z}^t$ and $\boldsymbol{\tau}^t$ are fixed, we compute $\mathbf{x}^{t+1}$ at iteration t by minimizing

$$\min_{\mathbf{x}} \ \|\mathbf{y}-\mathbf{x}\|_2^2 + \mu\,\mathbf{x}^\top \mathcal{L}_{\boldsymbol{\theta}}^{B}\mathbf{x} + (\boldsymbol{\tau}^t)^\top(\mathbf{x}-\mathbf{z}^t) + \frac{\rho}{2}\|\mathbf{x}-\mathbf{z}^t\|_2^2. \tag{19}$$

(19) is convex and quadratic, and thus the solution can be obtained by solving the following linear system:

$$\left((1+\frac{\rho}{2})\mathbf{I} + \mu\mathcal{L}_{\boldsymbol{\theta}}^{B}\right)\mathbf{x}^{t+1} = \mathbf{y} - \frac{\boldsymbol{\tau}^t}{2} + \frac{\rho}{2}\mathbf{z}^t. \tag{20}$$

One way to compute solution $\mathbf{x}^{t+1}$ in (20) without matrix inverse is *conjugate gradient*[6] (CG) (Hestenes et al., 1952). However, unrolling CG into neural layers and tuning CG parameters, as done in Thuc et al. (2024), only optimizes a generic linear system solver.

Instead, we compute $\mathbf{x}^{t+1}$ via *Lanczos approximation* of graph filters (Susnjara et al., 2015), so that spectrum-aware parameters can be learned after Lanczos unrolling[7]. First, given polarity assignment $\boldsymbol{\beta}$ for balanced Laplacian $\mathcal{L}_{\boldsymbol{\theta}}^{B}$ satisfying (1), compute positive graph input $\mathbf{y}^+ = \mathbf{S}(\mathbf{y} - \frac{\boldsymbol{\tau}^t}{2} + \frac{\rho}{2}\mathbf{z}^t)$ for $\mathbf{S} = \mathrm{diag}(\boldsymbol{\beta})$. Then, compute positive graph Laplacian $\mathcal{L}_{\boldsymbol{\theta}}^{+} = \mathbf{S}\mathcal{L}_{\boldsymbol{\theta}}^{B}\mathbf{S}^{-1}$, so that positive graph filter output $\mathbf{x}^+$ for *low-pass* filter response[8] $r(\lambda) = (1 + \frac{\rho}{2} + \mu\lambda)^{-1}$ can be approximated as

$$\mathbf{x}^+ = r(\mathcal{L}_{\boldsymbol{\theta}}^{+})\,\mathbf{y}^+ \approx \left\|\mathbf{y}^+\right\|_2 \mathbf{V}_m\, r(\mathbf{H}_m)\,\mathbf{c}_1 \tag{21}$$

where $\mathbf{H}_m = \mathbf{V}_m^\top \mathcal{L}_{\boldsymbol{\theta}}^{+}\mathbf{V}_m$ is an $m \times m$ tridiagonal matrix, $\mathbf{V}_m$ are orthonormal Lanczos vectors, and $\mathbf{c}_1$ is the first canonical vector. $m \ll N$ is the number of Lanczos iterations. After obtaining $\mathbf{x}^+$ from (21), we recover solution $\mathbf{x}^{t+1} = \mathbf{S}\mathbf{x}^+$ from the corresponding positive graph output $\mathbf{x}^+$.

3.4.2 Solving for Auxiliary Variable $\mathbf{z}$

When $\mathbf{x}^{t+1}$ and $\boldsymbol{\tau}^t$ are fixed, at iteration t we compute $\mathbf{z}^{t+1}$ by minimizing

$$\min_{\mathbf{z}} \ \mu\,\mathbf{z}^\top \mathbf{F}_{\boldsymbol{\theta}}\mathbf{z} + (\boldsymbol{\tau}^t)^\top(\mathbf{x}^{t+1}-\mathbf{z}) + \frac{\rho}{2}\|\mathbf{x}^{t+1}-\mathbf{z}\|_2^2. \tag{22}$$

[5] Since $\mathbf{F}_{\boldsymbol{\theta}}$ contains only frustrated-edge entries, the diagonal load $\delta\mathbf{I}$ is absorbed into the balanced component $\mathcal{L}_{\boldsymbol{\theta}}^{B}$, while preserving the similarity transform $\mathcal{L}_{\boldsymbol{\theta}}^{+} = \mathbf{S}\mathcal{L}_{\boldsymbol{\theta}}^{B}\mathbf{S}^{-1}$ to a positive graph Laplacian.

[6] The complexity of CG solving a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is $\mathcal{O}(\mathrm{nnz}(\mathbf{A})\sqrt{\kappa(\mathbf{A})}\log\frac{1}{\epsilon})$, where $\mathrm{nnz}(\mathbf{A})$ is the number of non-zero entries in $\mathbf{A}$, $\kappa(\mathbf{A})$ is the condition number, and ϵ is the convergence threshold.

[7] See Appendix F for an explanation on why Lanczos filter approximation is advantageous over CG. See Appendix G for an explanation on why Lanczos is performed on balanced graph Laplacian $\mathcal{L}_{\boldsymbol{\theta}}^{B}$ instead of $\mathcal{L}_{\boldsymbol{\theta}}$.

[8] $r(\lambda)$ is a low-pass filter, since larger frequency means stronger attenuation, *i.e.*, $r(\lambda') < r(\lambda), \forall\lambda' > \lambda$.

Note here that $\mathbf{F}_{\boldsymbol{\theta}}$ is not necessarily PSD. However, assuming $\frac{\rho}{2}\mathbf{I} + \mu\mathbf{F}_{\boldsymbol{\theta}}$ is PD (ρ is typically large), (22) is convex and quadratic, and solution $\mathbf{z}$ can be obtained by solving a linear system:

$$\left(\frac{\rho}{2}\mathbf{I} + \mu\mathbf{F}_{\boldsymbol{\theta}}\right)\mathbf{z}^{t+1} = \frac{\boldsymbol{\tau}^t}{2} + \frac{\rho}{2}\mathbf{x}^{t+1}. \tag{23}$$

Solution $\mathbf{z}^{t+1}$ can also be computed from (23) without matrix inverse via CG.

3.4.3 Updating Lagrange Multiplier

Lagrange multiplier $\boldsymbol{\tau}$ can be updated in the usual ADMM manner (Boyd et al., 2011):

$$\boldsymbol{\tau}^{t+1} = \boldsymbol{\tau}^t + \rho\left(\mathbf{x}^{t+1} - \mathbf{z}^{t+1}\right). \tag{24}$$

4 Algorithm Unrolling & Depression Score Prediction

The alternating GEM algorithm designed for a pretext denoising task is unrolled into a feed-forward network for parameter tuning, as shown in Fig. 2. Each unrolled M-step constructs an initial signed graph $\mathcal{G}^s$ via (8), computes polarities $\boldsymbol{\beta}$ via (10), and regulates frustration via (11) in order to output a near-balanced graph Laplacian $\mathcal{L}_{\boldsymbol{\theta}}$. To construct initial $\mathcal{G}^s$, a shallow temporal CNN $F(\cdot)$ combines ROI signals and anatomical coordinates into node features, which are refined across blocks to parameterize signed edge weights. Each unrolled E-step computes primary variable $\mathbf{x}^{t+1}$ via Lanczos low-pass approximation in (21), computes auxiliary variable $\mathbf{z}^{t+1}$ by solving linear system (23) via CG, and updates the Lagrange multiplier τ^{t+1} in (24), in order to output a denoised signal $\mathbf{x}$. Parameters learned in the M-step are parameters of feature function $F(\cdot)$, metric matrix $\mathbf{M}$, switching point $d^\star$ and shape parameter ξ. Parameters learned in the E-step are parameters in the unrolled Lanczos (see Appendix H for details), parameters in the unrolled CG, and ADMM parameter ρ. The network is trained to minimize signal reconstruction MSE in a supervised manner.

Subsequently, we employ the pretrained denoising network for the downstream depression score prediction task. In particular, we adopt the denoising network for feature extraction, where the parameters and layer-specific graphs of the pretrained network remain frozen. Given the fMRI window of each subject, we extract two complementary feature sets from the denoising network: 1) ROI features $\mathbf{f}_i$ of the ith node that describe regional signal characteristics, which are extracted from the feature function $F(\cdot)$; 2) graph-guided features computed from each unrolled block, including inter-ROI feature distances $d_{i,j}$ in (8) and the magnitudes and alignment of the balanced and frustrated component responses:

$$\mathbf{R}_\ell^B = -\mu_\ell\left(\mathcal{L}_{\boldsymbol{\theta}}^{B,(\ell)} - \delta_\ell\mathbf{I}\right)\widehat{\mathbf{X}}^{(\ell)}, \quad \mathbf{R}_\ell^F = -\mu_\ell\mathbf{F}_{\boldsymbol{\theta}}^{(\ell)}\widehat{\mathbf{X}}^{(\ell)}, \tag{25}$$

where $\widehat{\mathbf{X}}^{(\ell)}$ is the denoised output of block ℓ. The response magnitudes quantify the strength of each component's action on the signal, while their alignment indicates whether the two actions reinforce or oppose each other. The graph-guided features are summarized within anatomical ROI groups and combined across blocks. Then we apply PCA to the ROI features and graph-guided features separately for dimensionality reduction. Based on these features, we train a linear predictor with BDI scores as targets, and take average of window-level predictions to obtain one BDI score for each subject. Please note that the PCA and the predictor are fitted using training subjects only.

5 Experiments

5.1 Experimental Setup

We predict subject-level BDI scores from resting-state fMRI, where BDI quantifies self-reported depressive symptom severity. Our cohort comprises 65 participants with available BDI scores (47 with depression and 18 healthy controls) from OpenNeuro ds002748 (Ray et al., 2021). Following XCP-D postprocessing of fMRIPrep outputs (Mehta et al., 2024), we retain time series from 94 AAL116 ROIs. Within each participant, ROI signals are z-score normalized over the full scan before segmentation into 64×94 (time $\times$ ROI) windows. Our model consists of three unrolled blocks, each combining graph learning with ADMM-based filtering. All blocks share a fixed 256-edge candidate support constructed from anatomical, spatial priors and ROI correlations. All experiments follow a leave-one-subject-out (LOSO) protocol and are conducted on an NVIDIA GeForce RTX 3090 GPU. See Appendix I for experiment implementation details.

Table 1: **Comparison with Baselines on BDI Prediction.** PCC and SCC denote Pearson's and Spearman's correlation coefficients. The full comparison is provided in the appendix.

Category	Method	Params.	MAE↓	RMSE↓	PCC↑	SCC↑
Non-graph Based	Tangent Ridge (Dadi et al., 2019)	4.4K	9.35	11.31	0.183	0.159
	FC-LASSO (Yamashita et al., 2021)	$\leq$ 4.4K	10.06	12.34	-0.189	-0.431
	Spectral DCM–PEB (Ray et al., 2021)	N/A	12.02	15.72	0.109	-0.285
Graph Based	BrainGNN (Li et al., 2021)	77.4K	11.73	14.20	-0.092	-0.115
	STAGIN (Kim et al., 2021)	894.7K	21.33	27.57	0.035	0.033
	BNT (Kan et al., 2022)	1.2M	9.37	11.49	0.263	0.263
	graphTRIP-BDI (Tolle et al., 2025)	31.0K	9.76	11.68	-0.289	-0.323
	FDSyn-GNN (Chen et al., 2026a)	723.0K	10.74	13.01	0.128	0.127
Graph-based	**Near-Balanced Signed Graph (Ours)**	1.9K	**8.72**	**10.32**	**0.439**	**0.453**

Table 2: **Ablation of graph structure.** Graph structures are evaluated on denoising and BDI prediction under the same 65-subject LOSO protocol.

Category	Configuration	Denoising		BDI prediction	
		MSE↓	$\triangle$SNR↑	MAE↓	PCC↑
Graph structure	Fixed FC graph	0.1643	1.781	11.28	0.083
	Positive graph	0.1504	2.217	9.41	0.127
	Full-Balanced signed graph	0.1442	2.400	13.79	-0.379
	Near-balanced signed graph	**0.1441**	**2.415**	**8.72**	**0.439**

5.2 Prediction Results

Our method achieves the lowest prediction errors and highest correlations among the methods in Table 1. MAE and RMSE reach 8.72 and 10.32, representing reductions of 6.7% and 8.8% relative to Tangent Ridge. PCC and SCC increase to 0.439 and 0.453, exceeding BNT by 0.176 and 0.190, respectively. These improvements reflect gains in both absolute score accuracy and the ordering of subjects by symptom severity. With approximately 1.9K parameters, the framework requires only about 0.16% of BNT's parameter count. The results support the utility of optimization-unrolled graph denoising for learning compact representations for BDI prediction.

5.3 Ablation Studies

Table 2 shows that the near-balanced graph outperforms fixed FC and positive graphs in both denoising and BDI prediction. Strict balance achieves comparable denoising MSE, yet substantially poorer prediction, with PCC dropping from 0.439 to -0.379. It also exhibits unstable edge weights and isolated nodes, indicating that low reconstruction error alone does not ensure a stable, transferable graph representation. These findings support allowing controlled frustration rather than enforcing balance when learning graph representations for the downstream task. Additional ablations on solver budgets, unrolling depth, and graph-filter strength etc. are provided in Appendix J.

6 Conclusion

For signals with strong anti-correlations that tend towards a near-balanced signed graph structure, we build a new generalized expectation maximization (GEM) framework that alternately learns a near-balanced graph (M-step) and denoises a signal given the learned graph (E-step). Specifically, in the M-step we construct a near-balanced graph using learned features and regulating graph frustration. Given a learned near-balanced graph, in the E-step we decompose the graph Laplacian regularization term into balanced and frustrated components, so that respective spectral low-pass filter and linear system solver can be alternately applied in an ADMM setup. Experimental results show that our proposal achieves SOTA depression score prediction while reducing parameters to fewer than 5% of SOTA schemes, outperforming previous positive and full-balanced signed graph models.

STATEMENT

We used generative AI tools for language polishing and assistance with repeated ablation experiment runs and parameter adjustments. The authors manually inspected the execution logs, verified the experimental outputs and reported results, and take responsibility for the final content of this work. The fMRI experiments use the publicly available OpenNeuro ds002748 dataset for secondary analysis; no identifiable patient information or other sensitive personal data are disclosed in this paper or the supplementary materials. An anonymized code package will be provided as supplementary material. Data preprocessing, model configuration, and training procedures are described in Appendix I, while evaluation protocols and ablation settings are documented in Appendix J.

REFERENCES

T. Antal, P. L. Krapivsky, and S. Redner. Dynamics of social balance on networks. *Phys. Rev. E*, 72 (3):036121, September 2005.

Jyotismita Barman, Mohammad Yusuf, Sandeep Kumar, and Tapan Kumar Gandhi. Enhancing depression diagnosis with augmented brain signal driven decorrelated graph neural networks. *Communications Medicine*, 6, 2026. doi: 10.1038/s43856-026-01395-y.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. In *Foundations and Trends in Optimization*, volume 3, no.1, pp. 1–122, 2011.

Jianghe Cai, Gene Cheung, and Fei Chen. Unrolling plug-and-play gradient graph laplacian regularizer for image restoration. *IEEE Transactions on Image Processing*, 35:1817–1831, 2026. doi: 10.1109/TIP.2025.3562425.

Zhihong Chen, Jiayi Peng, Xiaorui Han, Mengfan Wang, Jiang Wu, Xinhua Wei, Zhengze Gong, Dezhong Yao, Li Pu, and Hongmei Yan. An interpretable functional-dynamic synaptic graph neural network for major depressive disorder diagnosis from rs-fmri. *International Journal of Neural Systems*, 36(9):2650024, 2026a. doi: 10.1142/S0129065726500243.

Zhihong Chen, Jiang Wu, Li Pu, Shaoqing Li, Yuhao Zhang, Dezhong Yao, and Hongmei Yan. MAF-GNN: Graph neural network-based multi-atlas brain functional information fusion for major depressive disorder diagnosis with rs-fmri. *Information Processing & Management*, 63: 104524, 2026b. doi: 10.1016/j.ipm.2025.104524.

G. Cheung, E. Magli, Y. Tanaka, and M. Ng. Graph spectral image processing. In *Proceedings of the IEEE*, volume 106, no.5, pp. 907–930, May 2018.

Jeongwhan Choi, Hyowon Wi, Jayoung Kim, Yehjin Shin, Kookjin Lee, Nathaniel Trask, and Noseong Park. Graph convolutions enrich the self-attention in transformers! In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 52891–52936. Curran Associates, Inc., 2024. doi: 10.52202/079017-1676. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5eceb48c3bc8b5d936c05ff8e2ece65e-Paper-Conference.pdf.

Kamalaker Dadi, Mehdi Rahim, Alexandre Abraham, Darya Chyzhyk, Michael Milham, Bertrand Thirion, and Gael Varoquaux. Benchmarking functional connectome-based predictive models for resting-state fMRI. *NeuroImage*, 192:115–134, 2019. doi: 10.1016/j.neuroimage.2019.02.062.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

Fernando Diaz-Diaz, Karel Devriendt, and Renaud Lambiotte. Gremban expansion for signed networks: Algebraic and combinatorial foundations for community–faction detection. *arXiv preprint arXiv:2509.14193*, 2025. doi: 10.48550/arXiv.2509.14193. URL https://arxiv.org/abs/2509.14193.

Chinthaka Dinesh, Gene Cheung, Saghar Bagheri, and Ivan V. Bajić. Efficient signed graph sampling via balancing & Gershgorin disc perfect alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2330–2348, 2025. doi: 10.1109/TPAMI.2024.3524180.

Parham Eftekhar, Gene Cheung, Mingxiao Liu, H. Vicky Zhao, Xuejun Han, and Eugene Wen. Unrolling Lanczos for ideal low-pass graph filter approximation, 2026. arXiv:2609.29086.

Anna Gallo, Diego Garlaschelli, Renaud Lambiotte, Fabio Saracco, and Tiziano Squartini. Testing structural balance theories in heterogeneous signed networks. *Commun Phys*, 7(1):1–13, May 2024a. ISSN 2399-3650.

Anna Gallo, Diego Garlaschelli, and Tiziano Squartini. Assessing frustration in real-world signed networks: A statistical theory of balance. *Physical Review Research*, 6:L042065, 2024b. doi: 10.1103/PhysRevResearch.6.L042065.

G. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in the Mathematical Sciences)*. Johns Hopkins University Press, 2012.

Frank Harary. On the notion of balance of a signed graph. *Mich. Math. J.*, 2(2):143–146, 1953.

Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2nd edition, 2012. ISBN 978-0-521-54823-6.

Jiangping Hu and Wei Xing Zheng. Bipartite consensus for multi-agent systems on directed signed networks. In *52nd IEEE Conference on Decision and Control*, pp. 3451–3456, December 2013.

Vassilis Kalofolias. How to learn a graph from smooth signals. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 920–929, Cadiz, Spain, 09–11 May 2016. PMLR. URL https://proceedings.mlr.press/v51/kalofolias16.html.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Alec Kirkley, George T. Cantwell, and M. E. J. Newman. Balance in signed networks. *Physical Review E*, 99(1):012320, 2019. doi: 10.1103/PhysRevE.99.012320.

Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 23(2):517–541, 2001.

Andrew V. Knyazev. On spectral partitioning of signed graphs. In *Proceedings of the 2018 SIAM Workshop on Combinatorial Scientific Computing (CSC18)*, pp. 81–91. SIAM, 2018. doi: 10.1137/1.9781611975215.9.

Elizabeth Kresock, Bryan A. Dawkins, Henry Luttbeg, Yijie Li, Rayus Kuplicki, and Brett A. McKinney. Centrality nearest-neighbor projected-distance regression (C-NPDR) feature selection for correlation-based predictors with application to resting-state fMRI study of major depressive disorder. *PLOS ONE*, 20(3):e0319346, 2025. doi: 10.1371/journal.pone.0319346.

Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W. De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM)*, pp. 559–570, 2010. doi: 10.1137/1.9781611972801.49. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972801.49.

Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, et al. BrainGNN: Interpretable brain graph neural network for fMRI analysis. *Medical Image Analysis*, 74:102233, 2021. doi: 10.1016/j.media.2021.102233.

Xinyu Liang, Yunan Guo, Hanyue Zhang, Xiaotong Wang, Danian Li, Yujie Liu, Jianjia Zhang, Luping Zhou, and Shijun Qiu. Neuroimaging signatures and a deep learning modeling for early diagnosing and predicting non-pharmacological therapy success for subclinical depression comorbid sleep disorders in college students. *International Journal of Clinical and Health Psychology*, 24(4):100526, 2024. doi: 10.1016/j.ijchp.2024.100526.

Seth A. Marvel, Steven H. Strogatz, and Jon M. Kleinberg. Energy landscape of social balance. *Physical Review Letters*, 103:198701, 2009. doi: 10.1103/PhysRevLett.103.198701.

Kahini Mehta, Taylor Salo, Thomas J. Madison, et al. XCP-D: A robust pipeline for the post-processing of fMRI data. *Imaging Neuroscience*, 2:1–26, 2024. doi: 10.1162/imag_a_00257.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021. doi: 10.1109/MSP.2020.3016905.

A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. In *Proceedings of the IEEE*, volume 106, no.5, pp. 808–828, May 2018.

N. Parikh and S. Boyd. Proximal algorithms. In *Foundations and Trends in Optimization*, volume 1, no.3, pp. 123–231, 2013.

Ji Qi, Mingxiao Liu, Viet Ho Tam Thuc Do, Yuzhe Li, Zhuoshi Pan, Gene Cheung, and H. Vicky Zhao. Lightweight and interpretable transformer via unrolling of mixed graph algorithms for traffic forecast. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=UCLLoBmv6s.

Craig M. Rawlings and Noah E. Friedkin. The Structural Balance Theory of Sentiment Networks: Elaboration and Test. *American Journal of Sociology*, 123(2):510–548, September 2017.

Dipanjan Ray, Dmitry Bezmaternykh, Mikhail Mel'nikov, Karl J. Friston, and Moumita Das. Altered effective connectivity in sensorimotor cortices is a signature of severity and clinical course in depression. *Proceedings of the National Academy of Sciences*, 118(40):e2105730118, 2021. doi: 10.1073/pnas.2105730118.

Ana Susnjara, Nathanael Perraudin, Daniel Kressner, and Pierre Vandergheynst. Accelerated filtering on graphs using lanczos method, 2015. URL https://arxiv.org/abs/1509.04537.

Tam Thuc, Parham Eftekhar, Seyed Alireza Hosseini, Gene Cheung, and Philip A. Chou. Interpretable lightweight transformer via unrolling of learned graph smoothness priors. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 6393–6416. Curran Associates, Inc., 2024.

Hanna M. Tolle, Andrea I. Luppi, Timothy Lawn, Leor Roseman, David Nutt, Robin L. Carhart-Harris, and Pedro A. M. Mediano. Accurate and interpretable prediction of antidepressant treatment response from receptor-informed neuroimaging. *bioRxiv*, 2025. doi: 10.1101/2025.07.02.662710. Preprint; version 2 posted January 2026.

R. S. Varga. *Gershgorin and his circles*. Springer, 2004.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hyowon Wi, Jeongwhan Choi, and Noseong Park. Learning advanced self-attention for linear transformers in the singular value domain. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/730. URL https://doi.org/10.24963/ijcai.2025/730.

Ayumu Yamashita, Yuki Sakai, Takashi Yamada, et al. Common brain networks between major depressive-disorder diagnosis and symptoms of depression that are validated for independent cohorts. *Frontiers in Psychiatry*, 12:667881, 2021. doi: 10.3389/fpsyt.2021.667881.

Cheng Yang, Gene Cheung, and Wei Hu. Signed graph metric learning via Gershgorin disc perfect alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7219–7234, 2022. doi: 10.1109/TPAMI.2021.3091682.

Fengge Yao, Zimeng Yu, Jie Zhang, Bing Han, Yuehan Gao, Fengyu Dou, and Xiaotong Wen. Mapping depression correlated amygdala communication. *Brain Topography*, 39(5):84, 2026a. doi: 10.1007/s10548-026-01239-x.

Junyi Yao, Parham Eftekhar, Gene Cheung, Xujin Chris Liu, Yao Wang, and Wei Hu. Lightweight transformer for eeg classification via balanced signed graph algorithm unrolling. In *International Conference on Learning Representations (ICLR)*, 2026b. URL `https://openreview.net/forum?id=zxsLio384j`.

A Proof of Proposition 1

Proof. Write $\mathbf{R} := \mathbf{A} - \lambda_1 \mathbf{v}_1 \mathbf{v}_1^\top = \sum_{k\geq 2} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top$. Thus, spectral norm $\|\mathbf{R}\|_2 = \lambda_2 = \lambda_1 - \Delta$.

Flip condition. Fix (i,j), let $s = S^\star_{i,j} = \beta_i \beta_j$ and $p_{i,j} = |v_{1,i}||v_{1,j}|$. Then, $s\, v_{1,i} v_{1,j} = p_{i,j}$, so $sA_{i,j} = \lambda_1 p_{i,j} + sR_{i,j}$. If $(i,j) \in \mathcal{F}_\tau$, then $sA_{i,j} = -|A_{i,j}|$, hence

$$|R_{i,j}| \;\geq\; -sR_{i,j} \;=\; |A_{i,j}| + \lambda_1 p_{i,j} \;>\; \tau + \lambda_1 p_{i,j} \;\geq\; \tau. \tag{26}$$

Off-diagonal bound. For $i \neq j$, let $\mathbf{M}$ be the 2×2 principal submatrix of $\mathbf{R}$ on $\{i,j\}$, with diagonal $a, d \geq 0$ and off-diagonal $b = R_{ij}$. PSD gives $b^2 \leq ad$, and the Rayleigh quotient of $\mathbf{R}$ at $(\mathbf{e}_i \pm \mathbf{e}_j)/\sqrt{2}$ gives $\frac{a+d}{2} + |b| \leq \lambda_2$. Since $\frac{a+d}{2} \geq \sqrt{ad} \geq |b|$, we get $|R_{i,j}| \leq \lambda_2/2$.

(i) If $\tau \geq \lambda_2/2$, then (26) would force $|R_{i,j}| > \lambda_2/2$, a contradiction.

(ii) By (26), $\tau^2 |\mathcal{F}_\tau| \leq \sum_{i<j} R_{ij}^2 = \frac{1}{2}\big(\|\mathbf{R}\|_F^2 - \sum_i R_{ii}^2\big)$. Let $t = \operatorname{tr}\mathbf{R} = \sum_{k\geq 2}\lambda_k$. Then $\|\mathbf{R}\|_F^2 = \sum_{k\geq 2}\lambda_k^2 \leq \lambda_2 t$ and $\sum_i R_{ii}^2 \geq t^2/N$ by Cauchy–Schwarz, so $\sum_{i<j} R_{ij}^2 \leq \frac{1}{2}(\lambda_2 t - t^2/N) \leq N\lambda_2^2/8$, maximizing over t.

(iii) By (ii), $\#\{(i,j) : |R_{ij}| > \tau\} \leq \binom{N}{2}\varepsilon$ and $|\mathcal{F}_\tau| \leq \binom{N}{2}\varepsilon$. Let $\theta = \lambda_1 c^2/N > 2\tau$, so $\lambda_1 p_{i,j} \geq \theta$ for all pairs. If $|R_{i,j}| \leq \tau$, then $|A_{i,j}| \geq \lambda_1 p_{i,j} - |R_{i,j}| > \tau$, and so $(i,j) \in \mathcal{E}_\tau$. Hence, $|\mathcal{E}_\tau| \geq \binom{N}{2}(1-\varepsilon)$, and the claim follows. □

B Edge Weight assignment via a Mass-Spring Interpretation

Following the mass-spring interpretation of signed graphs, positive edge weights encode *attractive* spring interactions, while negative edge weights encode *repulsive* interactions (Knyazev, 2018). Denote by $d_{i,j} \geq 0$ the Mahalanobis distance between nodes i and j in a learned feature space, such as (8). We assume that nearby samples in feature space should exhibit net attraction, while sufficiently distant samples should exhibit net repulsion. This suggests a signed edge weight $w_{i,j}$ that decreases monotonically with distance and changes sign at a critical distance $d^\star$.

To formalize this intuition, suppose attractive and repulsive affinities decay exponentially with distance:

$$A(d) = e^{-d/\tau}, \qquad R(d) = e^{-(2d^\star - d)/\tau}, \tag{27}$$

where $\tau > 0$ controls the interaction scale. We define the effective interaction strength as the normalized difference between attraction and repulsion,

$$w(d) = \frac{A(d) - R(d)}{A(d) + R(d)}. \tag{28}$$

Substituting the affinity expressions yields

$$w(d) = \tanh\left(\frac{d^\star - d}{\tau}\right). \tag{29}$$

This interpretation provides a physical motivation for the chosen distance-to-weight mapping in (8). Parameter $d^\star$ denotes the distance at which attractive and repulsive effects balance, yielding $w(d^\star) = 0$, while τ controls the width of the transition region. The resulting function is bounded in $(-1, 1)$, treats attraction and repulsion symmetrically, and saturates for very small or very large distances, preventing a small number of extreme distances from dominating the signed graph structure.

C Unrolling of Graph Filter Algorithm into Transformer

Following Thuc et al. (2024), we show that a combination of a normalized graph learning module and a low-pass filter on the learned graph operates like a self-attention mechanism, and thus unrolling a learned graph filter algorithm is a transformer. We first review the classical self-attention mechanism (Vaswani et al., 2017), which has three fundamental matrices: *query* $\mathbf{Q}$, *key* $\mathbf{K}$, and *value* $\mathbf{V}$, each matrix in $\mathbb{R}^{M\times M}$. Given input embeddings $\mathbf{x}_i \in \mathbb{R}^M$ for each token i, first, the *affinity* score $a_{i,j}$ is computed using a scaled dot product. Then, the *attention weight* $w_{i,j}$ is obtained through a row-wise softmax:

$$a_{i,j} = (\mathbf{Q}\mathbf{x}_i)^\top (\mathbf{K}\mathbf{x}_j)/\sqrt{M}, \quad w_{i,j} = \frac{\exp(a_{i,j})}{\sum_l \exp(a_{i,l})}. \tag{30}$$

Finally, output embeddings $\mathbf{Y} \in \mathbb{R}^{N\times M}$ are computed as the weighted combinations of input embeddings $\mathbf{X} \in \mathbb{R}^{N\times M}$ (N tokens in rows, each $\mathbf{x}_i$ of $\mathbb{R}^M$), multiplied by $\mathbf{V}$ that mixes M channels:

$$\mathbf{Y} = \mathbf{W}(\mathbf{X})\,\mathbf{X}\,\mathbf{V}. \tag{31}$$

Note that the attention weight (adjacency) matrix $\mathbf{W}(\mathbf{X}) \in \mathbb{R}^{N\times N}$ is a function of the input embeddings $\mathbf{X}$. Written in this form, *self-attention can be interpreted as a graph filtering operation*[9] on a learned graph whose adjacency matrix is given by $\mathbf{W}(\mathbf{X})$. Since $\mathbf{W}(\mathbf{X})\,\mathbf{X}$ aggregates neighboring embeddings through weighted averaging, it exhibits a smoothing (low-pass) behavior.

Consider now a graph learning module that first computes *feature vector* $\mathbf{f}_i \in \mathbb{R}^K$ for each node i, then compute non-negative feature distance $d_{i,j}$ between samples i and j as the Mahalanobis distance given PSD metric matrix $\mathbf{M} \succeq 0$. Non-negative edge weight $w_{i,j} \in [0, 1]$ connecting nodes i and j is computed as the exponential of the negative distance, *i.e.*, $w_{i,j} = \exp(-d_{i,j})$:

$$d_{i,j} = (\mathbf{f}_i - \mathbf{f}_j)^\top \mathbf{M}(\mathbf{f}_i - \mathbf{f}_j), \quad \bar{w}_{i,j} = \frac{\exp(-d_{i,j})}{\sum_l \exp(-d_{i,l})} \tag{32}$$

where $\bar{w}_{i,j}$ is the normalized version of $w_{i,j}$, such that the sum of edge weights stemming from node i equals to 1. Examining (30) and (32), one can see that self-attention weight is like normalized graph edge weight, if one interprets affinity $a_{i,j}$ as negative distance $-d_{i,j}$. Combining with a low-pass filter $h(\mathbf{L}(\mathbf{X}))$ for signal-dependent Laplacian $\mathbf{L}(\mathbf{X})$, one can thus conclude that a combination of a normalized graph learning module that constructs $\mathbf{L}(\mathbf{X})$ and a low-pass filter $h(\cdot)$, *i.e.*, $h(\mathbf{L}(\mathbf{X}))\mathbf{X}$, is akin to self-attention $\mathbf{W}(\mathbf{X})\mathbf{X}$. (Channel mixing operator $\mathbf{V}$ can be optionally added in both cases.) Since a transformer is a concatenation of the self-attention mechanism in a sequence, we conclude that unrolling a learned graph filter algorithm results in a transformer-like network.

D Gremban Expansion

A Gremban expansion (Diaz-Diaz et al., 2025) lifts an *unbalanced* signed graph $\mathcal{G}$ of N nodes to a *balanced* signed graph $\mathcal{G}^B$ of $2N$ nodes via the following construction.

1. Replicate node set $\mathcal{V}$ in $\mathcal{G}$ into two sets $\mathcal{V}^+$ and $\mathcal{V}^-$ in $\mathcal{G}^B$.

[9] This graph filter interpretation is also suggested in Choi et al. (2024) and Wi et al. (2025). However, instead of reducing parameters via feature graph learning, their goal is to generalize the weighted average filter $\mathbf{W}(\mathbf{X})$ into more general filters, resulting in larger models with more parameters.

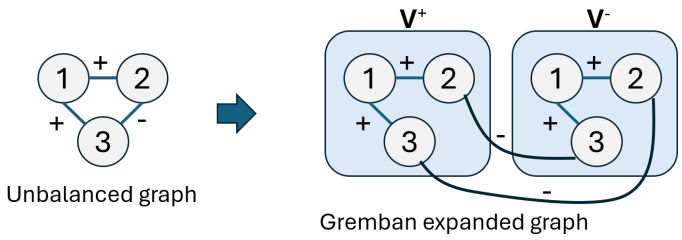


Figure 3: Example of a 3-node unbalanced signed graph $\mathcal{G}$ being converted to a 6-node Gremban expanded graph $\mathcal{G}^B$. $\mathcal{G}^B$ is balanced, since nodes within each of $\mathcal{V}^+$ and $\mathcal{V}^-$ are connected by positive edges only, while negative edges connect only nodes between $\mathcal{V}^+$ and $\mathcal{V}^-$.

2. For each *positive* edge $(i, j) \in \mathcal{E}$ in $\mathcal{G}$, draw two edges (i^+, j^+) and (i^-, j^-) connecting nodes *within* $\mathcal{V}^+$ and $\mathcal{V}^-$, respectively.

3. For each *negative* edge $(i, j) \in \mathcal{E}$ in $\mathcal{G}$, draw two edges (i^+, j^-) and (i^-, j^+) connecting nodes *across* sets $\mathcal{V}^+$ and $\mathcal{V}^-$.

See Fig. 3 for an example of a 3-node unbalanced signed graph $\mathcal{G}$ (with a cycle of one negative edge) that is converted to a 6-node Gremban expanded graph $\mathcal{G}^B$. Clearly $\mathcal{G}^B$ is balanced; nodes within $\mathcal{V}^+$ and $\mathcal{V}^-$ are connected by positive edges, and all negative edges connect nodes from $\mathcal{V}^+$ and $\mathcal{V}^-$. Thus, nodes in $\mathcal{V}^+$ and $\mathcal{V}^-$ can be assigned polarities $\beta_i = 1$ and -1, respectively, that satisfy (1).

One important property of the Gremban expanded graph $\mathcal{G}^B$ is that *if the original signed graph $\mathcal{G}$ is balanced, then $\mathcal{G}^B$ is disconnected*. Suppose we order nodes i in $\mathcal{G}$ with polarity $\beta_i = 1$ (subset $\mathcal{V}_1$ of size k) before nodes with $\beta_i = -1$ (subset $\mathcal{V}_{-1}$ of size $N - k$). Then, we can write the $N \times N$ adjacency matrix $\mathbf{W}$ for $\mathcal{G}$ as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1^+ & -\mathbf{W}^- \\ -(\mathbf{W}^-)^\top & \mathbf{W}_{-1}^+ \end{bmatrix} \tag{33}$$

where $\mathbf{W}_1^+ \in \mathbb{R}^{k \times k}$ and $\mathbf{W}_{-1}^+ \in \mathbb{R}^{(N-k)\times(N-k)}$ are symmetric adjacency sub-matrices connecting nodes within subset $\mathcal{V}_1$ and $\mathcal{V}_{-1}$ respectively, and $\mathbf{W}^- \in \mathbb{R}^{k\times(N-k)}$ connects k nodes in $\mathcal{V}_1$ to $N - k$ nodes in $\mathcal{V}_{-1}$. Note that $\mathbf{W}_1^+, \mathbf{W}_{-1}^+, \mathbf{W}^-$ are all non-negative matrices.

Next, we can write the $2N \times 2N$ adjacency matrix $\mathbf{W}^B$ for the expanded graph $\mathcal{G}^B$ as

$$\mathbf{W}^B = \begin{bmatrix} \mathbf{W}_1^+ & & & -\mathbf{W}^- \\ & \mathbf{W}_{-1}^+ & -(\mathbf{W}^-)^\top & \\ & -\mathbf{W}^- & \mathbf{W}_1^+ & \\ -(\mathbf{W}^-)^\top & & & \mathbf{W}_{-1}^+ \end{bmatrix}. \tag{34}$$

One can reorder entries in $\mathbf{W}^B$ into a *block diagonal* matrix, and thus $\mathcal{G}^B$ is disconnected.

From (34), we can deduce that a *minimum cut* procedure that removes edges and separates nodes in $\mathcal{G}^B$ into two symmetric sets, $\mathcal{V}_1^+ \cup \mathcal{V}_{-1}^-$ and $\mathcal{V}_{-1}^+ \cup \mathcal{V}_1^-$, would result in a polarity assignment $\boldsymbol{\beta}$ and a set of frustrated edges $\mathbf{F}$. However, the computation cost of operating on a $2N$-node Gremban expanded graph is still high. Hence, for polarity assignment $\boldsymbol{\beta}$ on an unbalanced signed graph, we compute instead the first eigenvector of the signed graph Laplacian $\mathbf{L}_{\boldsymbol{\theta}}^s$ instead via efficient LOBPCG, as written in (10).

E Choice of Laplacian Matrices for Signed Graphs

We discuss why the signed graph Laplacian $\mathbf{L}_{\boldsymbol{\theta}}^s$ is used over the combinatorial graph Laplacian $\mathbf{L}_{\boldsymbol{\theta}}$ for polarity assignment and frustration regulation in Section 3.3. Recall that the signed graph Laplacian is defined as $\mathbf{L}_{\boldsymbol{\theta}}^s \triangleq \mathbf{D}^s - \mathbf{W}$, where $\mathbf{D}^s$ is a diagonal absolute degree matrix with diagonal

entries $D^s_{i,i} = \sum_j |w_{i,j}|$. Thus, the quadratic Rayleigh quotient of $\mathbf{L}^s_{\boldsymbol{\theta}}$ for unit-norm $\mathbf{x}$ is

$$\mathbf{x}^\top \mathbf{L}^s_{\boldsymbol{\theta}} \mathbf{x} = \mathbf{x}^\top (\mathbf{D}^s - \mathbf{W})\mathbf{x} = \mathbf{x}^\top \mathbf{D}^s \mathbf{x} - \mathbf{x}^\top \mathbf{W} \mathbf{x} \tag{35}$$

$$= \sum_i D^s_{i,i} x_i^2 - \sum_{i,j} w_{i,j} x_i x_j \tag{36}$$

$$= \sum_i \sum_j |w_{i,j}| x_i^2 - \sum_i \sum_j w_{i,j} x_i x_j \tag{37}$$

$$= \frac{1}{2} \sum_i \sum_j |w_{i,j}| \left(x_i^2 + x_j^2 - 2\,\mathrm{sign}(w_{i,j}) x_i x_j \right) \tag{38}$$

$$= \frac{1}{2} \sum_i \sum_j |w_{i,j}| (x_i - \mathrm{sign}(w_{i,j}) x_j)^2. \tag{39}$$

(39) implies that signed graph Laplacian $\mathbf{L}^s_{\boldsymbol{\theta}}$ is PSD and $\mathbf{x}^\top \mathbf{L}^s_{\boldsymbol{\theta}} \mathbf{x} \geq 0, \forall \mathbf{x}$. In contrast, the combinatorial graph Laplacian $\mathbf{L}_{\boldsymbol{\theta}}$ can be indefinite if negative edges exist. This explains why $\mathbf{L}^s_{\boldsymbol{\theta}}$ is used in the frustration regulation objective in (11): to ensure that the minimization objective is non-negative. It also means that the first eigenvector—unit-norm vector that minimizes the Rayleigh quotient—is the solution to the relaxed polarity optimization (10).

F Lanczos Approximated Graph Filter versus Conjugate Gradient

A natural question is why we approximate the solution of (20) via Lanczos approximation of the corresponding graph spectral filter (Susnjara et al., 2015), rather than simply applying CG and, if parameter learning is desired, unrolling CG's iterations into neural layers as done in Thuc et al. (2024). We argue that, while both approaches are iterative, matrix-free, and capable of solving (20) exactly given a sufficient number of iterations, they are fundamentally different in what a *truncated* version of each method computes, and consequently, in what learned parameters within each method actually accomplish.

CG is a general-purpose iterative solver for *symmetric positive definite* (SPD) linear systems $\mathbf{A}\mathbf{x} = \mathbf{b}$ (Hestenes et al., 1952). It makes no assumptions on the origin, structure, or semantic meaning of $\mathbf{A}$ or $\mathbf{b}$. Hence, it is equally applicable whether $\mathbf{A}$ arises from a graph Laplacian or any other SPD operator. When T iterations of CG are unrolled into neural layers—with learnable per-iteration step sizes and momentum (conjugate direction) coefficients—the resulting learned parameters are tuned to accelerate numerical convergence of the residual $\|\mathbf{A}\mathbf{x}^{(t)} - \mathbf{b}\|_2$ toward zero for the specific family of matrices $\mathbf{A} = (1 + \rho/2)\mathbf{I} + \mu \mathcal{L}^B_{\boldsymbol{\theta}}$ encountered during training. This objective is entirely about convergence *speed and accuracy of the linear solver itself*. It carries no explicit connection to the GSP interpretation of the problem: the solution $\mathbf{x}^\star = \mathbf{A}^{-1}\mathbf{b}$ is the output of a low-pass graph filter $r(\lambda) = (1 + \rho/2 + \mu\lambda)^{-1}$ applied to the input signal $\mathbf{b}$. Two truncated-CG solutions with identical residual norm may differ arbitrarily in how well they approximate this filtering behavior across the graph spectrum, since residual norm and spectral filter fidelity are not the same quantity. In this sense, unrolling CG and learning parameters optimizes a proxy objective—numerical convergence—that deviates from the signal processing task the E-step is meant to perform.

In contrast, the Lanczos approximation in (21) is a specific method designed to approximate the action of a matrix function, $r(\mathcal{L}^B_{\boldsymbol{\theta}})\mathbf{b}$, on a vector $\mathbf{b}$ (Susnjara et al., 2015). When we truncate the Lanczos process at $m \ll N$ iterations, we are *directly* approximating the frequency response $r(\lambda)$ of the low-pass filter over the spectrum of $\mathcal{L}^B_{\boldsymbol{\theta}}$ using a degree-m polynomial (implicitly, via the Ritz values and vectors of the tridiagonal matrix $\mathbf{H}_m$) that is most accurate for the frequency components of $\mathbf{b}$ that dominate its Krylov subspace expansion. The truncation error incurred by stopping at m Lanczos steps is hence not a numerical solver error, but a *filter approximation error*: it directly quantifies how faithfully the truncated filter $r_m(\lambda)$ reproduces the low-pass response $r(\lambda)$ across the relevant portion of the graph spectrum. Because this error has a clear signal-processing interpretation, any learnable parameters introduced when unrolling the Lanczos steps are tuned to directly improve low-pass filtering fidelity, which is the quantity we care about for denoising performance in the E-step. There is no analogous, task-aligned interpretation available for parameters unrolled within CG.

This distinction is not a matter of computational efficiency, since both truncated CG and truncated Lanczos are matrix-free and share comparable per-iteration cost when applied to sparse graph operators. It is also not a matter of correctness, since both methods converge to the identical exact solution $\mathbf{x}^\star$ of (20) as the iteration count increases. It is instead a matter of what a *fixed, finite-depth, learnable* truncation of each method represents. Learning CG parameters optimizes convergence of a generic linear solver toward a target that happens to be a graph low-pass filter, but the learned parameters themselves have no filtering semantics. Learning Lanczos parameters optimizes the fidelity of an explicit low-pass graph spectral filter, and the learned parameters are therefore directly optimized for filtering performance. Since the entire premise of the observation model in (15) is that $\mathbf{x}^{t+1}$ should be recovered as the output of smoothing $\mathbf{y}$ (adjusted by the ADMM auxiliary terms) with respect to the graph structure encoded in $\mathcal{L}_{\boldsymbol{\theta}}^B$, the Lanczos-based construction is the appropriate architectural choice.

G Lanczos Approximation on Balanced Signed Graph

Another question is why spectral filtering in the E-step is applied only to the balanced component $\mathcal{L}_{\boldsymbol{\theta}}^B$, instead of directly filtering with the full near-balanced Laplacian

$$\mathcal{L}_{\boldsymbol{\theta}} = \mathcal{L}_{\boldsymbol{\theta}}^B + \mathbf{F}_{\boldsymbol{\theta}}. \tag{40}$$

From a numerical linear algebra perspective, there is no fundamental obstacle to applying Lanczos approximation directly to $\mathcal{L}_{\boldsymbol{\theta}}$. Lanczos is a general-purpose Krylov-subspace method for approximating matrix functions $g(\mathbf{L})\mathbf{b}$ when $\mathbf{L}$ is symmetric, and its convergence theory does not require that $\mathbf{L}$ be a graph Laplacian, nor that it admit any manifold interpretation. Hence, the motivation for filtering using $\mathcal{L}_{\boldsymbol{\theta}}^B$ instead of $\mathcal{L}_{\boldsymbol{\theta}}$ is not algorithmic but statistical.

Our basic premise, as articulated in Proposition 1, is that the underlying signal statistics are *near-balanced*. Specifically, we assume that the dominant dependency structure can be explained by a balanced signed graph, while a relatively small number of frustrated edges capture interactions that cannot be represented by a globally consistent polarity assignment. Writing $\mathcal{L}_{\boldsymbol{\theta}} = \mathcal{L}_{\boldsymbol{\theta}}^B + \mathbf{F}_{\boldsymbol{\theta}}$, the balanced component $\mathcal{L}_{\boldsymbol{\theta}}^B$ represents the principal geometric structure of the data, while $\mathbf{F}_{\boldsymbol{\theta}}$ captures departures from that structure. This distinction is important because balanced signed graphs admit a one-to-one correspondence with positive graphs (see Section 2.3). Specifically, given diagonal polarity matrix $\mathbf{S} = \mathrm{diag}(\boldsymbol{\beta})$,

$$\mathcal{L}_{\boldsymbol{\theta}}^+ = \mathbf{S}\mathcal{L}_{\boldsymbol{\theta}}^B\mathbf{S}^{-1} \tag{41}$$

is the Laplacian of a corresponding positive graph. Consequently, $\mathcal{L}_{\boldsymbol{\theta}}^B$ inherits the standard graph-smoothness interpretation associated with positive graph Laplacians. Under the usual assumptions of graph manifold learning, positive graph Laplacians constructed from nonnegative similarity kernels converge to the Laplace–Beltrami operator of an underlying manifold. The eigenvectors of $\mathcal{L}_{\boldsymbol{\theta}}^B$ can therefore be viewed as discrete analogues of manifold harmonics, ordered according to increasing oscillation with respect to the intrinsic geometry.

The significance of this observation is the following. If the latent signal is genuinely smooth with respect to the underlying manifold, then its spectral representation in the eigenbasis of $\mathcal{L}_{\boldsymbol{\theta}}^B$ is expected to be concentrated at low frequencies:

$$\mathbf{x} = \sum_k \alpha_k \mathbf{v}_k^{(B)}, \qquad |\alpha_k| \to 0 \quad \text{rapidly as } k \to \infty. \tag{42}$$

In other words, smooth signals admit a compact representation in the spectral coordinates defined by $\mathcal{L}_{\boldsymbol{\theta}}^B$. Lanczos approximation is signal-dependent.

Although worst-case error bounds are determined by polynomial approximation of the filter response, practical performance also depends on how the input signal energy is distributed across the operator spectrum. When the signal energy is concentrated in low-frequency eigenvectors, a low-dimensional Krylov subspace captures a larger fraction of the signal energy, and a truncated Lanczos approximation can achieve high filtering accuracy with relatively few iterations. In contrast, the frustration component $\mathbf{F}_\theta$ does not generally admit a positive-graph representation and therefore lacks a corresponding diffusion or manifold interpretation. Note that we do not view these frustrated edges as measurement noise; they model non-negligible non-manifold interactions that

cannot be explained by the dominant balanced structure. Examples include antagonistic relationships, long-range dependencies, or subject-specific deviations from the principal geometry.

Because such interactions do not define a meaningful notion of smoothness, incorporating them directly into the spectral operator may contaminate the frequency coordinates used for filtering. For this reason, we apply spectral filtering only with respect to $\mathcal{L}_{\boldsymbol{\theta}}^{B}$, which defines the dominant notion of smoothness, while treating the frustration component $\mathbf{F}_{\boldsymbol{\theta}}$ separately through the auxiliary ADMM update. This separation preserves a geometrically meaningful frequency interpretation for the filtering stage while still allowing frustrated interactions to influence the final solution.

H Unrolling the Lanczos Algorithm

We unroll m iterations of the Lanczos algorithm into m neural layers for data-driven parameter tuning, in order to better approximate low-pass filter response $r(\lambda) = (1 + \frac{\rho}{2} + \mu\lambda)^{-1}$. We first overview the Lanczos algorithm (Chapter 10, Golub & Loan (2012)). Given a unit-norm initial vector $\mathbf{v}_1$ and a real symmetric matrix $\mathbf{L}$, the m-th *Krylov space* is defined as

$$\mathcal{K}_m(\mathbf{L}, \mathbf{v}_1) \triangleq \operatorname{span}(\{\mathbf{v}_1, \mathbf{L}\mathbf{v}_1, \ldots, \mathbf{L}^{m-1}\mathbf{v}_1\}). \tag{43}$$

The Lanczos algorithm iteratively computes m orthonormal vectors $\mathbf{V}_m = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$—called the *Lanczos vectors*—that span the Krylov space. Specifically, $\mathbf{L}$ and $\mathbf{V}_m$ are related by

$$\mathbf{V}_m^\top \mathbf{L} \mathbf{V}_m = \mathbf{H}_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & & \\ & \beta_2 & \alpha_3 & \ddots & & \\ & & \ddots & \ddots & \beta_{m-2} & \\ & & & \beta_{m-2} & \alpha_{m-1} & \beta_{m-1} \\ 0 & & & & \beta_{m-1} & \alpha_m \end{bmatrix} \tag{44}$$

where $\mathbf{H}_m$ is an $m \times m$ symmetric tridiagonal matrix. In iterative algebraic form, we can write

$$\mathbf{w}_j = \mathbf{L}\mathbf{v}_j, \quad \alpha_j = \mathbf{v}_j^\top \mathbf{w}_j \tag{45}$$

$$\mathbf{r}_j = \mathbf{w}_j - \alpha_j \mathbf{v}_j - \beta_{j-1}\mathbf{v}_{j-1} \tag{46}$$

$$\beta_j = \|\mathbf{r}_j\|, \quad \mathbf{v}_{j+1} = \mathbf{r}_j / \beta_j \tag{47}$$

where (46) can be interpreted as an orthogonalization step, so that $\mathbf{r}_j$ (and subsequent normalized $\mathbf{v}_{j+1}$) is orthogonal to previously computed $\mathbf{v}_j, \mathbf{v}_{j-1}, \ldots, \mathbf{v}_1$, akin to the *Gram-Schmidt procedure* (Chapter 5, Golub & Loan (2012)).

To introduce parameters in Lanczos for data-driven learning, instead of (46) we relax the orthogonality condition and add learnable parameters, $\gamma_j^{(1)}$ and $\gamma_j^{(2)}$, at each step j:

$$\mathbf{r}_j = \mathbf{w}_j - \gamma_j^{(1)} \alpha_j \mathbf{v}_j - \gamma_j^{(2)} \beta_{j-1} \mathbf{v}_{j-1}. \tag{48}$$

The resulting vectors $\mathbf{v}_j$ are generally not mutually orthogonal. Consequently, the tridiagonal matrix $\mathbf{H}_m$ is no longer symmetric:

$$\mathbf{H}_m = \begin{bmatrix} \gamma_1^{(1)}\alpha_1 & \gamma_2^{(2)}\beta_1 & & & & 0 \\ \beta_1 & \gamma_2^{(1)}\alpha_2 & \gamma_3^{(2)}\beta_2 & & & \\ & \beta_2 & \ddots & \ddots & & \\ & & \ddots & \gamma_{m-1}^{(1)}\alpha_{m-1} & \gamma_m^{(2)}\beta_{m-1} & \\ 0 & & & \beta_{m-1} & \gamma_m^{(1)}\alpha_m & \end{bmatrix}. \tag{49}$$

To ensure $\mathbf{H}_m$ nonetheless has real-valued eigenvalues—so that $r(\mathbf{H}_m)$ in (21) is also real-valued—we impose constraints $\gamma_j^{(2)} > 0, \forall j$. See Lemma 1 in Eftekhar et al. (2026) for details. To train $\gamma_j^{(1)}$'s and $\gamma_j^{(2)}$'s, we can employ the following loss function

$$l_{\text{loss}} = \left\| r(\mathcal{L}_{\boldsymbol{\theta}}^{+})\mathbf{y}^{+} - \|\mathbf{y}^{+}\|_2 \mathbf{V}_m r(\mathbf{H}_m)\mathbf{c}_1 \right\|_2 \tag{50}$$

to enhance signal-dependent low-pass filter response approximation for input signal $\mathbf{y}^{+}$.

I Experimental Details

I.1 Data Preparation and Denoising Targets

The source derivatives were generated with fMRIPrep 21.0.2 and postprocessed with XCP-D 26.1.1 using the 36-parameter confound model, despiking, a 0.01–0.08 Hz temporal passband, and an FD censoring threshold of 0.3 mm. The network reads the native XCP-D AAL116 ROI time-series files, retaining the fixed ordering of 94 ROIs with valid measurements. It does not reconstruct ROI signals from randomly sampled voxels. Voxel-subset augmentation and its view-consistency loss are not enabled in the current experiment.

Each ROI is z-score normalized independently within each subject over the complete available scan, using the population standard deviation. This precedes window extraction; there is no additional window-level z-scoring or cross-subject signal normalization. Windows contain 64 time points and use stride two, with the final valid window included even when its starting index is not on the stride grid. Each window has shape 64×94 with one signal channel. Near-constant ROIs trigger an explicit validation failure rather than silent zero filling. All windows of an outer test subject remain outside upstream and downstream fitting.

The denoising reference is the preprocessed ROI window before synthetic corruption, not noise-free physiological ground truth. Independent Gaussian noise with standard deviation $\sigma = 0.5$ is added once before the first block and redrawn at every optimizer update. No additional noise is injected between blocks. A single label-free denoiser is trained per LOSO fold; BDI scores and diagnosis labels are not inputs to its encoder, graph construction, or filters.

I.2 Candidate Graph and Graph-Learning Configuration

Candidate selection uses only the 64 training subjects in each outer fold. The 256-edge undirected support includes 44 literature-informed connections, 41 available bilateral homologous connections, and up to 32 additional negatively correlated pairs. The latter require a Fisher-averaged training correlation of at most -0.2 and negative correlations in at least 75% of training subjects; eligible pairs are ranked by negative correlation strength without a spatial restriction. The remaining edges minimize total distance between atlas ROI centers, subject to connectivity and a minimum degree of two. Reference weights are training correlations restricted to this support. All three blocks share the same candidate support, which remains fixed during training but may differ across LOSO folds.

ROI features $\mathbf{f}_i$ are normalized to unit ℓ_2 norm before metric projection. Each block has an independent 16×16 metric factor, normalized such that the induced PSD matrix $\mathbf{M}$ has trace 16. Pairwise distances are averaged with equal subject weight. After frustration regulation, we normalize the adjacency as $\mathbf{W} \leftarrow (\mathbf{D}^s)^{-1/2}\mathbf{W}(\mathbf{D}^s)^{-1/2}$, with absolute degrees floored at 10^{-4}, and rebuild the Laplacians from these weights.

The implementation retains the smoothness–degree–frustration tradeoff in (11), but penalizes relative rather than absolute frustration:

$$\upsilon \frac{\sum_{(i,j)\in\mathcal{E}_F} m_{i,j}}{\sum_{(i,j)\in\mathcal{E}} m_{i,j}}, \qquad m_{i,j} \geq 0. \tag{51}$$

We use a log-degree coefficient of 2 and omit the upper bound $m_{i,j} \leq |w^o_{i,j}|$. Each block performs one projected-gradient update on frustrated-edge magnitudes, holding edge signs and balanced-edge magnitudes fixed. The coefficient υ is calibrated once per block and fold to yield an initial penalty gradient of 0.025 with respect to each frustrated-edge magnitude, then held fixed. The step size follows the graph optimizer's learning-rate schedule.

The switching point $d^\star$ and shape parameter ξ are calibrated from training-population distance statistics subject to a sign-ratio constraint, rather than learned by backpropagation. Polarity estimation uses dense spectral initialization followed by local discrete refinement. Updates are considered at epoch boundaries through epoch ten, require three consecutive agreeing proposals and a weighted-frustration improvement test, and change at most approximately 5% of nodes per accepted update. Polarities are subsequently frozen, while threshold calibration and trainable graph parameters remain adaptive.

I.3 Encoder and Filtering Blocks

The entry CNN uses bias-free convolutions with widths $1 \to 8 \to 1$, kernel sizes $(5, 3)$, and $\tanh$ activations. Each stage applies stride-two downsampling with the fixed kernel $(1, 4, 6, 4, 1)/16$, producing 16-dimensional features. Atlas coordinates, scaled by $1/100$, generate first-stage channel gains through a bias-free $3 \to 8$ map followed by $1 + 0.5 \tanh(\cdot)$. The two inter-block residual GNN modules use separate $16 \to 8$ maps for self, positive-neighbor, and negative-neighbor features, followed by $\tanh$ and an $8 \to 16$ projection.

The temporal kernels have lengths $(5, 5, 3)$ and contain 52 trainable parameters in total. Each block generates ROI-specific coefficients through a shared affine map of coordinates. Kernels have unit ℓ_2 norm, identity initialization, and replicated boundary padding. Tangent projection followed by norm restoration preserves total energy over time and ROIs within each window; no update-norm cap is imposed.

We use eight ADMM iterations, Lanczos order six, and at most 12 CG iterations, with relative and absolute CG tolerances of 10^{-6} and 10^{-8}. ADMM initializes $\mathbf{x} = \mathbf{z} = \mathbf{y}$ and $\boldsymbol{\tau} = 0$ for each block, with separate Krylov sequences for individual right-hand sides. The initial value of ρ is 0.5. The fixed layer-specific strengths $\mu = (0.05, 0.04, 0.025)$ replace the noise-derived setting $\mu = \sigma^2/2$, which equals 0.125 at $\sigma = 0.5$.

Diagonal loading checks both the full and balanced operators. For unshifted Laplacians $\mathbf{L}_{\boldsymbol{\theta}}$ and $\mathbf{L}^B_{\boldsymbol{\theta}}$, we set

$$\delta = \max\{0, -\lambda_{\min}(\mathbf{L}_{\boldsymbol{\theta}}), -\lambda_{\min}(\mathbf{L}^B_{\boldsymbol{\theta}})\}, \qquad \mathcal{L}_{\boldsymbol{\theta}} = \mathbf{L}_{\boldsymbol{\theta}} + \delta\mathbf{I}, \quad \mathcal{L}^B_{\boldsymbol{\theta}} = \mathbf{L}^B_{\boldsymbol{\theta}} + \delta\mathbf{I}. \tag{52}$$

Eigenvalues are computed by dense symmetric eigendecomposition. The shift leaves $\mathbf{F}_{\boldsymbol{\theta}}$ unchanged and preserves $\mathcal{L}_{\boldsymbol{\theta}} = \mathcal{L}^B_{\boldsymbol{\theta}} + \mathbf{F}_{\boldsymbol{\theta}}$. A numerical safeguard increases the effective ρ when necessary to ensure a Gershgorin lower bound of 10^{-5} for the frustrated linear system, using the same effective value in all ADMM updates. Both systems receive diagonal jitter 10^{-5}; reference solves include the same corrections.

I.4 Training Objective and Optimization

Each update uses one window from each of the 64 training subjects. Uncorrupted windows supply the graph-learning population, while independently corrupted copies enter the filtering branch. Epoch-end assessment and final graph caching use the full training-population reference, excluding the held-out subject.

Training minimizes final-output reconstruction MSE plus $0.5\mathcal{L}_{\mathrm{GLR}}$, where

$$\mathcal{L}_{\mathrm{GLR}} = \frac{1}{3N^2} \sum_{\ell=1}^{3} \mathrm{tr}\big(\mathbf{L}^s_{\boldsymbol{\theta}_\ell} \mathbf{C}_\ell\big), \qquad \mathbf{C}_\ell = \mathbb{E}[\mathbf{x}_\ell \mathbf{x}_\ell^\top], \quad N = 94. \tag{53}$$

Here, $\mathbf{L}^s_{\boldsymbol{\theta}_\ell}$ is constructed from the normalized adjacency, and $\mathbf{C}_\ell$ averages second moments over the corresponding block's training-population signals and time points. The second moments are detached, while the graph remains differentiable. No BDI supervision, classification, contrastive, or view-consistency loss is used upstream.

AdamW runs for 30 epochs with 19 subject-balanced updates per epoch, totaling 570 updates per fold. Balanced repeat queues supplement subjects with fewer windows. Following one warm-up epoch, cosine schedules reduce the encoder/graph learning rate from 10^{-3} to 10^{-6} by epoch ten and the temporal-filter rate from 3×10^{-3} to 3×10^{-5} by epoch twenty. Weight decay is 10^{-4} for the entry encoder and temporal kernels and zero for the remaining graph parameters. The global gradient norm is clipped at 1. Evaluation uses the prespecified epoch-30 checkpoint. Stability diagnostics do not trigger early stopping or certify optimality.

J Additional Ablations and Experimental Analysis

J.1 Full BDI Prediction Comparison

Table 3 extends the representative comparison in Table 1 to a broader set of brain graph learning methods under the same 65-subject leave-one-subject-out (LOSO) protocol.

Table 3: **Full Comparison on BDI Score Prediction.** Results are reported as mean ± sample standard deviation over five random seeds under the 65-subject LOSO protocol.

Method	Params.	MAE↓	RMSE↓	PCC↑	SCC↑
Non-graph-based Methods					
Tangent Ridge (Dadi et al., 2019)	4.4K	9.35 ± 0.18	11.31 ± 0.24	0.183 ± 0.026	0.159 ± 0.033
FC-LASSO (Yamashita et al., 2021)	$\leq$ 4.4K	10.06 ± 0.23	12.34 ± 0.31	-0.189 ± 0.041	-0.431 ± 0.052
C-NPDR + Ridge (Kresock et al., 2025)	≈ 65	10.12 ± 0.27	12.32 ± 0.36	-0.235 ± 0.049	-0.170 ± 0.056
Amygdala FC/EC–SVR (Yao et al., 2026a)	≈ 65	10.44 ± 0.22	12.85 ± 0.32	-0.018 ± 0.036	0.058 ± 0.043
Spectral DCM–PEB (Ray et al., 2021)	N/A	12.02 ± 0.42	15.72 ± 0.58	0.109 ± 0.059	-0.285 ± 0.067
Graph-based Methods					
BrainGNN (Li et al., 2021)	77.4K	11.73 ± 0.64	14.20 ± 0.87	-0.092 ± 0.091	-0.115 ± 0.103
STAGIN (Kim et al., 2021)	894.7K	21.33 ± 1.38	27.57 ± 1.84	0.035 ± 0.106	0.033 ± 0.112
BNT (Kan et al., 2022)	1.2M	9.37 ± 0.43	11.49 ± 0.56	0.263 ± 0.071	0.263 ± 0.079
HFBN-BDI (Liang et al., 2024)	280.4K	13.27 ± 0.78	17.34 ± 1.05	0.002 ± 0.083	-0.048 ± 0.094
MAF-GNN (Chen et al., 2026b)	411.2K	10.19 ± 0.48	12.15 ± 0.65	-0.053 ± 0.076	-0.049 ± 0.084
graphTRIP-BDI (Tolle et al., 2025)	31.0K	9.76 ± 0.36	11.68 ± 0.49	-0.289 ± 0.067	-0.323 ± 0.074
BrainADNet (Barman et al., 2026)	135.7K	10.11 ± 0.51	12.34 ± 0.69	0.062 ± 0.080	0.148 ± 0.089
FDSyn-GNN (Chen et al., 2026a)	723.0K	10.74 ± 0.57	13.01 ± 0.76	0.128 ± 0.078	0.127 ± 0.086
Near-Balanced Signed Graph (Ours)	1.9K	$\mathbf{8.72} \pm 0.58$	$\mathbf{10.32} \pm 0.72$	$\mathbf{0.439} \pm 0.068$	$\mathbf{0.453} \pm 0.075$

Baseline overview. The non-graph-based baselines predict BDI from explicit connectivity features or estimated connectivity parameters. Tangent Ridge (Dadi et al., 2019) combines tangent-space connectivity representations with ridge regression, while FC-LASSO (Yamashita et al., 2021) applies sparse linear regression to functional connectivity features. C-NPDR + Ridge (Kresock et al., 2025) introduces centrality-based feature selection before regression, and Amygdala FC/EC–SVR (Yao et al., 2026a) uses amygdala-centered functional and effective connectivity features with support vector regression. Spectral DCM–PEB (Ray et al., 2021) instead estimates directed effective connectivity and relates these parameters to BDI through parametric empirical Bayes.

The graph-based baselines learn representations of regional interactions and brain network organization. BrainGNN (Li et al., 2021) uses region-aware graph convolutions and pooling, BNT (Kan et al., 2022) combines attention with clustering-based readout, and HFBN-BDI (Liang et al., 2024) models hierarchical functional brain networks. STAGIN (Kim et al., 2021) captures dynamic connectivity through spatio-temporal attention, while FDSyn-GNN (Chen et al., 2026a) combines temporal signal encoding with connection-aware graph updates. MAF-GNN (Chen et al., 2026b) fuses complementary information across brain atlases, graphTRIP-BDI (Tolle et al., 2025) incorporates receptor-informed neuroimaging representations, and BrainADNet (Barman et al., 2026) combines signal augmentation with graph convolutions and feature decorrelation.

Prediction analysis. Consistent with the main-text comparison, our method achieves the best reported scores across all four metrics: an MAE of 8.72, an RMSE of 10.32, a PCC of 0.439, and an SCC of 0.453. Relative to Tangent Ridge, the strongest baseline in prediction error, our method reduces MAE and RMSE by 6.7% and 8.8%, respectively. Compared with BNT, which achieves the highest baseline correlations, it improves PCC by 0.176 and SCC by 0.190. The expanded comparison therefore supports improved absolute score estimation and subject ranking under the evaluated LOSO protocol. These prediction scores describe the complete pipeline; the reported 1.9K parameters refer only to the upstream denoiser and exclude the downstream prediction head.

J.2 Ablation Protocol and Metrics

We evaluate graph structure, filtering implementation, solver budgets, unrolling depth, frustration and GLR regularization, and graph-filter strength under the same 65-subject LOSO protocol. Unless varied, the reference uses three blocks, ADMM/Lanczos/CG budgets $(8, 6, 12)$, an external GLR coefficient $\lambda = 0.5$, and filter strengths $\boldsymbol{\mu} = (0.05, 0.04, 0.025)$. ROI preprocessing, Gaussian corruption ($\sigma = 0.5$), and subject-balanced training remain unchanged.

For each held-out subject s, let $E_{\mathrm{in},s}$ and $E_{\mathrm{out},s}$ denote input and reconstruction MSEs averaged over three windows and three paired noise realizations per window. With $S = 65$, we report

$$\mathrm{MSE} = \frac{1}{S}\sum_{s=1}^{S} E_{\mathrm{out},s}, \qquad \Delta\mathrm{SNR} = \frac{1}{S}\sum_{s=1}^{S} 10\log_{10}\frac{E_{\mathrm{in},s}}{E_{\mathrm{out},s}}. \tag{54}$$

$\Delta\mathrm{SNR}$ is expressed in dB. For budget and depth comparisons, latency covers the complete nine-view denoising evaluation, batching the three windows for each noise realization. Timing includes temporal filtering and solver updates, with weights, cached graph operators, and corrupted inputs preloaded; training, graph learning, disk I/O, and downstream prediction are excluded. On the same NVIDIA RTX 3090 GPU, we use ten warm-up passes and 30 synchronized repetitions, average runtimes within and across folds, and report latency relative to the reference configuration.

Budget sweeps reuse the frozen reference checkpoint and cached operators, with CG stopping tolerances fixed. Other variants are trained independently for 30 epochs using paired seeds, windows, and noise schedules, with the final checkpoint specified in advance. No outer-test results are used for model selection. Parameter counts refer only to the upstream denoiser.

BDI MAE and PCC supplement the graph-structure, depth, frustration, and GLR comparisons. A subject-balanced, window-level linear readout is fitted to frozen representations, and window predictions are averaged per subject. All readout transforms and regularization choices use training subjects only. Inner readout validation reuses the outer-training denoiser and is therefore not fully nested end-to-end validation.

J.3 Filter Iteration Budgets

Table 4 presents a sensitivity analysis around $(8, 6, 12)$, varying one inference budget at a time while fixing the remaining budgets, model parameters, graph operators, noise inputs, and CG stopping tolerances. Reducing ADMM iterations from eight to four, Lanczos order from six to four, or the CG cap from twelve to eight increases MSE by 6.6%, 4.3%, and 2.5%, respectively, with corresponding reductions in $\Delta\mathrm{SNR}$. Increasing budgets beyond the reference yields negligible quality gains (less than 0.03% in MSE) while increasing latency by 13–94%, illustrating diminishing returns from additional computation.

J.4 Number of Unrolled Blocks

Table 5 compares independently trained unrolling depths at fixed inner budgets $(8, 6, 12)$. Three blocks achieve the lowest denoising MSE of 0.14407 and the highest $\Delta\mathrm{SNR}$ of 2.415, while also yielding the best BDI prediction performance (MAE: 8.72; PCC: 0.439). Reducing the depth to one or two blocks increases denoising MSE by 25.7% and 4.2%, respectively. Increasing the depth to four or six blocks raises MSE by 4.5% and 12.8%, increases latency by approximately 34% and 95%, and worsens both BDI metrics. Among the evaluated depths, three blocks provide the best performance; additional refinement stages increase computational cost without improving reconstruction or downstream prediction.

J.5 Graph-Filter Strength

Table 6 reports a sensitivity analysis of $\boldsymbol{\mu} = c(0.05, 0.04, 0.025)$, with all other training and solver settings fixed. Relative to $c = 1$, reducing the multiplier to 0.75 and 0.5 increases MSE by 6.8% and 20.4%, respectively. Stronger filtering produces larger degradation: at $c = 1.5$ and $c = 2$, MSE increases by 27.9% and 58.4%, while $\Delta\mathrm{SNR}$ decreases from 2.415 to 1.343 and 0.419. The asymmetric degradation is consistent with insufficient noise suppression under weak filtering and greater smoothing bias under excessive filtering.

Table 4: **Effect of inner filter iteration budgets.** One budget is varied at a time around $(8, 6, 12)$ using the frozen reference checkpoint. Latency is normalized to the reference configuration.

ADMM	Lanczos	CG cap	MSE↓	ΔSNR↑	Relative latency↓
(a) ADMM iterations					
2	6	12	0.17382	1.600	0.27
4	6	12	0.15356	2.138	0.51
8	**6**	**12**	0.14407	2.415	1.00
16	6	12	0.14404	2.416	1.94
(b) Lanczos order					
8	2	12	0.16725	1.767	0.76
8	4	12	0.15028	2.232	0.88
8	**6**	**12**	0.14407	2.415	1.00
8	8	12	0.14406	2.415	1.13
8	12	12	0.14405	2.416	1.52
(c) Maximum CG iterations					
8	6	1	0.18864	1.244	0.68
8	6	2	0.16237	1.896	0.76
8	6	4	0.15091	2.214	0.84
8	6	8	0.14768	2.308	0.92
8	**6**	**12**	0.14407	2.415	1.00
8	6	24	0.14406	2.415	1.36

Table 5: **Effect of unrolling depth.** Each depth is trained independently at fixed inner budgets $(8, 6, 12)$. Params. counts the upstream denoiser, and latency is normalized to the three-block reference.

Blocks	Params.	Denoising			BDI prediction	
		MSE↓	ΔSNR↑	Relative latency↓	MAE↓	PCC↑
1	364	0.18113	1.421	0.34	10.10	−0.074
2	1,152	0.15019	2.234	0.66	9.85	−0.219
3 (reference)	1,932	**0.14407**	**2.415**	1.00	**8.72**	**0.439**
4	2,712	0.15062	2.222	1.34	9.97	0.131
6	4,272	0.16248	1.893	1.95	10.28	−0.094

Table 6: **Sensitivity to graph-filter strength.** A common multiplier scales the reference strength schedule, preserving its relative proportions.

Multiplier c	$\boldsymbol{\mu}$	MSE↓	ΔSNR↑
0.5	$(0.025, 0.020, 0.0125)$	0.17351	1.608
0.75	$(0.0375, 0.030, 0.01875)$	0.15382	2.129
1 (reference)	$(0.05, 0.04, 0.025)$	**0.14407**	**2.415**
1.5	$(0.075, 0.060, 0.0375)$	0.18430	1.343
2	$(0.10, 0.08, 0.05)$	0.22819	0.419