BAND-MASKED SPECTRAL AUTOENCODING FOR SELF-SUPERVISED GRAPH REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised graph representation learning is currently dominated by contrastive methods relying on engineered augmentations and masked autoencoders that reconstruct spatially masked features or edges. We introduce Band-Masked Spectral Autoencoding (BMSAE), a self-supervised framework that instead masks and reconstructs structured components of graph signals in the spectral domain. BMSAE decomposes node features into spectral bands using a fixed Meyer-wavelet Parseval frame, masks a subset of these bands, and predicts them from the remaining ones. Crucially, Parseval tightness yields an exact synthesis relation. This allows masked-band prediction error to bound reconstruction error in the original feature space and the change in a Lipschitz loss evaluated on the reconstruction. Because the spectral transform is applied independently to each graph, applying the node-level objective to a disjoint union is equivalent to applying it independently to each graph, while segment pooling yields graph-level representations. Experiments show that BMSAE is competitive with or improves on strong contrastive and masked-autoencoder baselines on graph-level benchmarks. For node classification, its strongest results occur on heterophilic graphs, where BMSAE outperforms competing methods on most of the evaluated benchmarks.

1 INTRODUCTION

Self-supervised graph representation learning is commonly based on either view agreement or masked reconstruction. Contrastive and teacher–student methods learn invariance across graph views generated by operations such as edge dropping, feature masking, or subgraph sampling. Their performance can therefore depend strongly on whether the chosen transformations preserve task-relevant information. Graph autoencoders reconstruct graph structure (Kipf & Welling, 2016), while masked graph autoencoders reconstruct masked node attributes or structural information (Hou et al., 2022; 2023). Coordinate-wise masking does not explicitly exploit the graph-frequency organization of node-feature signals.

We investigate masked autoencoding in the graph spectral domain. Given a graph signal, a fixed Meyer-wavelet Parseval frame can decompose it into frequency bands while retaining exact synthesis: the original signal can be recovered by Parseval synthesis of the band coefficients. The proposed method masks a nonempty proper subset of bands and predicts the held-out bands from the visible ones through a joint encoder and band-specific decoders; the main-comparison configuration uses uniform single-band masking, the Gate encoder, and linear decoders. Unlike coordinate-wise masking, the prediction target is an operator-defined component of the graph signal. Through Parseval synthesis, small masked-band prediction error implies small reconstruction error in the original feature space. We call this construction **Band-Masked Spectral Autoencoding (BMSAE)**.[1]

Its design uses a fixed, non-learned Meyer-wavelet Parseval frame, an encoder over the visible bands, and band-specific decoders for the held-out bands. The objective and reconstruction bound accommodate different encoder and decoder architectures; the gated encoder used in the main experiments and linear decoders are experimental choices. The theoretical results concern the ideal Parseval

[1] Code: `https://anonymous.4open.science/r/bmsae-2C03/`

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.

frame; in implementation, we approximate its continuous, piecewise smooth filter responses with Chebyshev recursions, avoiding eigendecomposition. We empirically check robustness to this approximation in Appendix C. Our contributions are:

- **Band-masked spectral autoencoding.** We introduce BMSAE, which predicts held-out graph-frequency bands from the visible bands of a fixed Meyer-wavelet Parseval frame. A joint encoder and band-specific decoders predict the held-out bands; the main comparison tables use the Gate encoder and linear decoders, while Appendix C reports independently selected graph-classification BMSAE variants with MLP and Attention encoders.
- **Parseval reconstruction guarantee.** For the ideal Parseval frame, masked-band prediction error bounds original-feature reconstruction error with a unit constant, and hence the change in a fixed predictor's Lipschitz composite loss on reconstructed features. We also bound departures from tightness and empirically assess the Chebyshev approximation. The guarantee is architecture-agnostic, so it applies to every BMSAE encoder and decoder variant we evaluate.
- **Evaluation across graph- and node-level tasks.** Under the reported frozen-probe protocols, BMSAE achieves higher mean scores than all evaluated baselines on five of six graph-classification datasets. It also achieves the highest mean score among the compared methods on four of five heterophilic node-classification datasets.

2 Related work

Contrastive and teacher–student graph SSL. Deep Graph Infomax and InfoGraph (Veličković et al., 2019; Sun et al., 2020) maximize agreement between local and global representations, while GraphCL and MVGRL (You et al., 2020; Hassani & Khasahmadi, 2020) contrast representations from augmented graph views. CCA-SSG (Zhang et al., 2021) and Graph Barlow Twins (Bielak et al., 2022) replace explicit negatives with feature-decorrelation objectives. Non-contrastive teacher–student methods such as BGRL (Thakoor et al., 2022) avoid negative pairs, but still learn through agreement between augmented views. These methods learn view agreement or invariance to perturbed inputs; BMSAE instead learns by predicting withheld spectral bands from the visible bands of a fixed Meyer-wavelet Parseval frame.

Generative / masked graph SSL. Graph autoencoders (Kipf & Welling, 2016) reconstruct graph structure, while GraphMAE and GraphMAE2 (Hou et al., 2022; 2023) mask node features in the spatial domain; S2GAE (Tan et al., 2023) masks edges. Whereas these methods reconstruct spatially masked attributes or edges, BMSAE withholds one or more complete spectral bands across the graph and predicts their coefficient matrices from the visible bands.

Spectral graph signal processing. Graph wavelets and tight frames (Hammond et al., 2011; Leonardi & Van De Ville, 2013; Shuman et al., 2013) provide the signal-processing tools on which BMSAE builds. Our contribution is not the frame itself but the spectral-band prediction task built on it; Parseval tightness supplies the exact adjoint synthesis that our analysis uses to relate band-prediction error to feature-reconstruction error.

Spectral graph SSL. Prior spectral SSL methods use graph spectra or wavelets to construct contrastive views, spectral augmentations, or learnable filtering operators. For example, Sp2GCL and related methods contrast spectral or wavelet-derived representations (Bo et al., 2023; Liu et al., 2024; Wan et al., 2024), while other approaches perturb frequency components to generate augmented views (Liu et al., 2022; Lin et al., 2023; Zhang et al., 2023). These cited methods use spectral structure within view-based objectives. BMSAE instead uses a fixed Meyer-wavelet Parseval frame in a generative pretext task: it masks a nonempty proper subset of spectral bands and predicts their coefficient matrices from the visible bands.

In short, BMSAE's distinguishing element is its pretext task: predicting complete, withheld frequency bands of a fixed Meyer-wavelet Parseval frame from the visible bands. The Parseval analysis (Section 5) provides a reconstruction guarantee for this task.

3 PRELIMINARIES

We introduce the three ingredients used throughout the paper: the graph Laplacian for defining spectral bands, Chebyshev recursions for evaluating them efficiently, and Parseval frames for reconstructing features exactly from those bands.

Graph Laplacian and spectral bands. For an undirected graph with adjacency A, degree matrix D, and node-feature matrix $X \in \mathbb{R}^{n\times p}$ (n nodes and p input features), we use the symmetric normalized Laplacian $L_{\mathrm{sym}} = I - D^{-1/2}AD^{-1/2}$, setting isolated-node entries of $D^{-1/2}$ to zero. Its spectrum lies in $[0, \lambda_{\max}] \subseteq [0, 2]$. If $L_{\mathrm{sym}} = U\Lambda U^\top$, a scalar response ψ defines the spectral filter $\psi(L_{\mathrm{sym}}) = U\psi(\Lambda)U^\top$ and the corresponding filtered features $\psi(L_{\mathrm{sym}})X$. Lower graph frequencies describe degree-normalized signals that vary smoothly across edges, whereas higher frequencies describe more rapid edgewise variation.

Parseval bands. The fixed Meyer bank $\{\psi_j\}_{j=1}^V$ produces overlapping spectral-band coefficients $Z_j = \psi_j(L_{\mathrm{sym}})X$. Its analysis and synthesis operators are

$$T : X \mapsto (\psi_j(L_{\mathrm{sym}})X)_{j=1}^V, \qquad T^*(Y_1, \ldots, Y_V) = \sum_{j=1}^V \psi_j(L_{\mathrm{sym}})Y_j.$$

The ideal bank is Parseval:

$$T^*T = I, \qquad X = \sum_{j=1}^V \psi_j(L_{\mathrm{sym}})Z_j.$$

Our ideal bank satisfies $\sum_j \psi_j(\mu)^2 = 1$ throughout the chosen spectral interval. Thus Z_j is a band coefficient, not itself a synthesis contribution; the latter is $\psi_j(L_{\mathrm{sym}})Z_j$. Appendix B.2.1 gives the fixed raised-cosine Meyer responses and the Linear/Sqrt band-boundary construction; Appendix A.5 contrasts this choice with general non-tight frames.

Chebyshev evaluation. We approximate each continuous, piecewise smooth response without eigendecomposition using a degree-K Chebyshev polynomial on the shared interval $[0, 2.02]$. The sparse recurrence is computed once per graph and reused by all V filters, avoiding dense $n \times n$ spectral operators. The identities above and our main theorem concern the ideal filters; the finite-K implementation is approximate. Its numerical robustness is evaluated in Appendix C, and its conditional reconstruction extension is in Appendix A.2.

4 BAND-MASKED SPECTRAL AUTOENCODING

BMSAE learns graph representations through a self-supervised masked-band prediction task. Given node features and a fixed Parseval analysis frame, it computes spectral-band coefficients, withholds a nonempty proper subset of complete bands, encodes the visible coefficients, and predicts each withheld coefficient with a band-specific decoder. Thus the framework is defined by fixed-frame analysis, band withholding, visible-band encoding, and masked-coefficient prediction; encoder depth and decoder form are experimental configuration choices. The ideal-frame reconstruction result is stated in Section 5.

BMSAE decomposes each graph's features into fixed spectral bands $Z_j = \psi_j(L_{\mathrm{sym}})X$ using the Parseval frame from Section 3; it learns neither the transform nor its band boundaries. BMSAE-Linear and BMSAE-Sqrt prescribe different fixed center-spacing rules: for V band centers spanning $[0, 2.02]$, Linear places the centers uniformly in the Laplacian spectral coordinate, whereas Sqrt uses squared normalized positions and therefore places more centers at lower graph frequencies. They are tuned independently, so their selected configurations may also differ in latent width, gate depth, learning rate, and weight decay. For readability, the method equations below use these ideal coefficients. The implementation instead substitutes their degree-K Chebyshev approximations $\widetilde{Z}_j$, as shown in Algorithm 1.

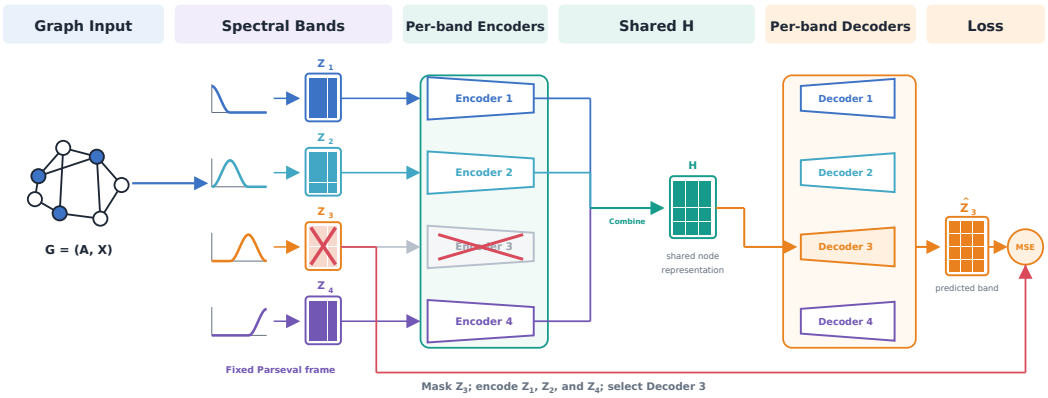

Figure 1: BMSAE illustrated using the gated per-band encoder and linear decoders employed in the main experiments. In this four-band example, Z_3 is withheld from the encoder input and reconstructed from the visible bands Z_1, Z_2, and Z_4. Algorithm 1 describes the general framework: the encoder may jointly process visible bands without separate per-band branches, and the masked set may contain multiple bands.

4.1 Masked-band objective

Let $V \geq 2$, let M be a mask satisfying $\varnothing \neq M \subsetneq \{1, \ldots, V\}$, let $\bar{M} = \{1, \ldots, V\} \setminus M$ be the visible bands, and let q be a distribution over such masks. General BMSAE encodes the indexed visible coefficients jointly,

$$H_M = f_\theta\big(\{(i, Z_i) : i \in \bar{M}\}\big), \qquad \hat{Z}_j = g_{\phi,j}(H_M), \quad j \in M.$$

Thus the encoder is informed of which bands are visible. For one graph, $H_M \in \mathbb{R}^{n \times d_{\text{lat}}}$; for a stacked collection, it has one row per stacked node. It may contain band-specific branches and a fusion into H_M, so the notation does not imply that every band shares encoder parameters. Encoder-family comparisons are reported in Appendix C. For per-graph coefficients of shape $n \times p$ (or their stacked disjoint-union analogue), BMSAE minimizes the expected masked-coefficient loss

$$\mathcal{L}_{\text{BMSAE}}(\theta, \phi) = \mathbb{E}_{M \sim q}\left[\frac{1}{|M| np} \sum_{j \in M} \left\| \hat{Z}_j - Z_j \right\|_{\text{F}}^2\right].$$

The main-comparison Gate configuration is the special case with q uniform over single-band masks: one sampled target band is excluded from the encoder input for every node, rather than being implemented by zeroing selected entries. It uses a band-specific MLP h_{θ_i} for every visible band and dataset-level scalar gate logits a_i. Its encoder is

$$H_M = \sum_{i \in \bar{M}} \alpha_i(M)\, h_{\theta_i}(Z_i), \qquad \alpha_i(M) = \frac{\exp(a_i)}{\sum_{r \in \bar{M}} \exp(a_r)}.$$

The gate is renormalized over precisely the visible bands for each mask, while its learned scalar logits are shared across all nodes and graphs processed by a trained model. The logits are initialized to zero, so the gate weights are uniform over the visible bands at initialization. For $V{=}2$ with a single masked band, exactly one band is visible, so its renormalized weight is one and the reconstruction loss provides no gradient to either gate logit. The training representation is then that visible band's MLP output $h_{\theta_i}(Z_i)$, from which the linear decoder produces the reconstruction; learned gate weighting plays no role. The depth of each band-specific MLP, one or two, is selected as part of the experimental configuration. The framework permits different band-specific decoder architectures; the current main experiments use one $\text{Linear}(d_{\text{lat}} \to p)$ layer for each target band. The reconstruction bound depends on the resulting band-prediction errors, not on decoder linearity.

Algorithm 1 BMSAE training and embedding extraction. The main-comparison configuration uses uniform single-band masking, the Gate encoder, and linear target-band decoders.

Require: graphs $\{(A^{(g)}, X^{(g)})\}_{g=1}^{G}$ with $X^{(g)} \in \mathbb{R}^{n_g \times p}$; $V \geq 2$; mask distribution q supported on nonempty proper band subsets; encoder f_θ; target-band decoders $\{g_{\phi,j}\}_{j=1}^{V}$; Chebyshev order K

form $L_{\mathrm{sym}}^{(g)}$ from each $A^{(g)}$
construct fixed Linear- or Sqrt-spaced Meyer responses $\{\psi_j\}_{j=1}^{V}$ and degree-K approximations $\{\widetilde{\psi}_{j,K}\}_{j=1}^{V}$ ▷ Appendix B.2.1
for each graph g and band j **do**
    $\widetilde{Z}_j^{(g)} \leftarrow \widetilde{\psi}_{j,K}(L_{\mathrm{sym}}^{(g)})X^{(g)}$
end for
stack graphwise bands as $\widetilde{Z}_j$; retain graph segment ids; set $N \leftarrow \sum_{g=1}^{G} n_g$
while not converged **do** ▷ main experiments: fixed 400 epochs
    sample one shared $M \sim q$ for all graph blocks ▷ main experiments: uniform single-band mask
    $H_M \leftarrow f_\theta(\{(i, \widetilde{Z}_i) : i \notin M\})$
    for each $j \in M$ **do**
        $\widehat{\widetilde{Z}}_j \leftarrow g_{\phi,j}(H_M)$
    end for
    $\mathcal{L} \leftarrow \frac{1}{|M|Np} \sum_{j \in M} \left\| \widehat{\widetilde{Z}}_j - \widetilde{Z}_j \right\|_{\mathrm{F}}^2$
    update (θ, ϕ) using $\nabla_{\theta,\phi}\mathcal{L}$
end while
freeze f_θ and compute $H_{\mathrm{eval}} \leftarrow f_\theta(\{(i, \widetilde{Z}_i)\}_{i=1}^{V})$
return node rows of H_{eval}, or graph embeddings $[\mathrm{mean} \| \mathrm{sum} \| \mathrm{max}]$ within each segment

4.2 Node and graph embedding extraction

For a collection of graphs with compatible node-feature dimension, BMSAE filters each graph with its own Laplacian and retains a graph-segment identifier when stacking node rows. Thus spectral computation remains independent within each graph. A shared sampled mask is applied to all stacked graphs. For the Gate encoder used in the main experiments, node representations are computed without cross-graph mixing, using parameters shared across graphs. Let n_g be the number of nodes in graph g and $N = \sum_g n_g$. For any fixed mask and parameter values, the stacked objective is

$$\mathcal{L}_{\mathrm{union}}(M) = \sum_g \frac{n_g}{N} \mathcal{L}_g(M).$$

Thus, disjoint-union equivalence refers to evaluating and aggregating per-graph losses with shared parameters, rather than equally weighting graphs or training a separate model for each graph. Appendix A.3 gives the corresponding direct-sum reconstruction result.

After training, we freeze the encoder and provide every indexed band,

$$H_{\mathrm{eval}} = f_\theta\big(\{(i, Z_i)\}_{i=1}^{V}\big).$$

Training therefore encodes incomplete visible-band sets, while inference uses all bands; the decoders are not used for read-out. For the gated implementation, the evaluation weights are renormalized over all V bands. Node embeddings are rows of H_{eval}. For graph classification, we pool latent rows within each graph segment with $\rho = \mathrm{mean} \,\|\, \mathrm{sum} \,\|\, \mathrm{max}$. Self-supervised training uses no labels; labels are used only for the downstream probe and model selection.

5 Theory

This section establishes the main ideal-frame reconstruction guarantee: small masked-band prediction error controls reconstruction error in node-feature space through Parseval synthesis. Proofs of

the prediction-loss statement and the extensions to Chebyshev filters and graph collections appear in Appendix A.

5.1 IDEAL-FRAME RECONSTRUCTION

For the ideal Parseval bank from Section 3, write $Z_j = \psi_j(L_{\mathrm{sym}})X$ and let M be the masked bands, with visible complement U. Its analysis operator is $T : X \mapsto (\psi_j(L_{\mathrm{sym}})X)_{j=1}^{V}$ and satisfies $T^*T = I$. Parseval tightness makes adjoint synthesis non-expansive: it cannot amplify the total stacked coefficient-prediction error. Given predictions $\hat{Z}_j$ for $j \in M$, synthesize

$$\hat{X} = \sum_{j \in U} \psi_j(L_{\mathrm{sym}})Z_j + \sum_{j \in M} \psi_j(L_{\mathrm{sym}})\hat{Z}_j.$$

Theorem 1 (Parseval-consistent reconstruction bound)**.** *For arbitrary masked-band predictions,*

$$\left\|\hat{X} - X\right\|_{\mathrm{F}}^2 \;\le\; \sum_{j \in M} \left\|\hat{Z}_j - Z_j\right\|_{\mathrm{F}}^2. \tag{1}$$

Consequently, if $\left\|\hat{Z}_j - Z_j\right\|_{\mathrm{F}} \le \varepsilon$ *for every* $j \in M$*, then* $\left\|\hat{X} - X\right\|_{\mathrm{F}} \le \varepsilon\sqrt{|M|}$.

Proof. Let $\Delta_j = \hat{Z}_j - Z_j$ for $j \in M$ and $\Delta_j = 0$ otherwise. The visible terms cancel, so $\hat{X} - X = T^*\Delta$. Parseval tightness gives $\|T^*\|_{\mathrm{op}} = 1$, and hence $\left\|\hat{X} - X\right\|_{\mathrm{F}}^2 \le \|\Delta\|_{\mathrm{F}}^2 = \sum_{j \in M} \left\|\hat{Z}_j - Z_j\right\|_{\mathrm{F}}^2$. The uniform consequence follows by bounding each summand by ε^2. □

Theorem 1 depends only on the ideal Parseval identity and the resulting coefficient-prediction errors, not on encoder depth, parameter sharing, or decoder linearity. It therefore applies to any architecture producing the stated predictions, without implying equal empirical errors across architectures.

Connection to the masked-band loss. For the ideal coefficients, the per-mask normalized loss is

$$\mathcal{L}_M := \frac{1}{|M|np} \sum_{j \in M} \left\|\hat{Z}_j - Z_j\right\|_{\mathrm{F}}^2.$$

Thus

$$\frac{\left\|\hat{X} - X\right\|_{\mathrm{F}}^2}{np} \le |M|\mathcal{L}_M. \tag{2}$$

For the reported single-band masks, the normalized ideal feature-reconstruction error is directly bounded by the masked-band MSE.

Corollary 1 (Prediction-loss stability under reconstruction)**.** *Let* x_v *and* $\hat{x}_v$ *denote row* v *of* X *and* $\hat{X}$*, and fix labels* $y_1, \ldots, y_n$ *and a scoring rule* r*. Define*

$$\widehat{R}_X(r) := \frac{1}{n} \sum_{v=1}^{n} \ell\big(r(x_v), y_v\big).$$

If $x \mapsto \ell(r(x), y)$ *is* L*-Lipschitz uniformly in* y*, then, for the same fixed rule* r *evaluated on* X *and* $\hat{X}$*,*

$$\left|\widehat{R}_{\hat{X}}(r) - \widehat{R}_X(r)\right| \;\le\; \frac{L}{\sqrt{n}} \left\|\hat{X} - X\right\|_{\mathrm{F}} \;\le\; L\sqrt{p|M|\mathcal{L}_M}. \tag{3}$$

This is an empirical ideal-frame statement for a fixed graph and rule r; it neither establishes population risk nor analyzes refitting the rule after reconstruction. The Lipschitz condition is on the composite loss, so it does not by itself control discontinuous metrics such as accuracy or ROC-AUC.

Appendix A.2 gives the approximation-aware Chebyshev bound: finite-order filtering adds a reconstruction residual controlled, conditionally, by the filter bank's tightness defect. Appendix A.3 gives the disjoint-union and pooled reconstructed-feature extensions.

Table 1: Graph-classification accuracy (%) under frozen linear probing. All deep baselines are reevaluated under the common protocol. For learned methods, values are mean ± population standard deviation across five encoder seeds, with accuracy averaged over the fixed ten folds for each seed. Deterministic references (Raw+LR, WL) report a single CV10 mean. V denotes the selected number of bands. The Δ row gives a descriptive best-of-two BMSAE comparison with the strongest displayed non-BMSAE result, including Raw+LR. Bold marks the highest mean per column, excluding Raw+LR. See Appendix B for experimental details.

Method	REDDIT-B.	REDDIT-M.	COLLAB	PROTEINS	D&D	NCI1
Shallow / non-deep references						
Raw+LR	77.0	55.0	72.7	73.9	77.6	66.1
WL (Shervashidze et al., 2011)	72.4	49.2	76.3	74.5	76.4	74.5
Contrastive self-supervised methods						
InfoGraph (Sun et al., 2020)	91.8±0.1	55.3±1.2	75.4±0.3	73.1±0.8	75.2±1.1	77.7±0.5
GraphCL (You et al., 2020)	90.9±0.4	55.0±0.4	76.2±0.3	70.9±1.1	74.0±1.3	**78.8±0.4**
Sp2GCL (Bo et al., 2023)	91.2±0.4	55.5±0.4	74.6±0.7	71.3±0.5	73.7±1.1	76.2±0.5
Masked / generative self-supervised methods						
GraphMAE (Hou et al., 2022)	91.0±0.4	54.9±0.4	77.1±0.4	69.6±1.0	72.2±0.5	75.2±1.9
S2GAE (Tan et al., 2023)	92.0±0.3	56.0±0.2	76.6±0.4	72.4±0.8	74.5±0.9	77.3±0.3
Spectral masked autoencoding						
BMSAE-Linear (ours)	93.1±0.3 (V=4)	**57.0±0.2** (V=2)	**77.7±0.4** (V=2)	**76.3±0.3** (V=2)	**78.6±0.6** (V=2)	76.8±0.2 (V=6)
BMSAE-Sqrt (ours)	**93.3±0.1** (V=3)	**57.0±0.2** (V=2)	**77.7±0.4** (V=2)	**76.3±0.3** (V=2)	**78.6±0.6** (V=2)	75.8±0.8 (V=4)
Δ *best BMSAE − best baseline*	+1.3	+1.0	+0.6	+1.8	+1.0	−2.0

6 EXPERIMENTS

6.1 EVALUATION SETUP

Evaluation setup. We evaluate frozen representations on six TU graph-classification datasets and five heterophilic node-classification benchmarks. Encoders are trained without labels, and downstream linear probes are fit using the labeled training partitions. Graph classification uses fixed stratified ten-fold evaluation; node classification uses the ten official benchmark splits. Dataset descriptions, preprocessing, and task-specific evaluation procedures are provided in Appendix B.

Methods compared. We evaluate the Linear and Sqrt variants defined in Section 4 against contrastive, masked/generative, spectral, and shallow baselines. Shared BMSAE settings and task-specific selection procedures are specified in Appendix B.2; baseline implementations and search spaces are documented in Appendix B.3.

6.2 GRAPH CLASSIFICATION

Graph-level results. The independently selected BMSAE variants attain the highest displayed mean among methods other than Raw+LR on REDDIT-BINARY (BMSAE-Sqrt: 93.3 ± 0.1), REDDIT-MULTI-5K (both variants: 57.0 ± 0.2), COLLAB (both variants: 77.7 ± 0.4), PROTEINS (both variants: 76.3 ± 0.3), and D&D (both variants: 78.6 ± 0.6). On NCI1, GraphCL has the highest mean among the evaluated deep baselines (78.8 ± 0.4 versus the best BMSAE result of 76.8 ± 0.2).

Selected spectral resolution. Both variants select $V=2$ on four of six graph datasets: REDDIT-MULTI-5K, COLLAB, PROTEINS, and D&D. This is the coarsest decomposition admitted by the objective, using two broad, overlapping responses emphasizing lower and higher frequencies. With single-band masking, the pretext task becomes reciprocal prediction: each update encodes one band and predicts the other. Because Linear and Sqrt spacing yield the same $V=2$ bank, their identical results on these datasets reflect the shared selected configuration and provide no comparison between spacing rules. The gate remains fixed at its zero initialization, assigning unit weight to the visible band during training and equal weights to the two band-specific MLP outputs at inference. These results show that a simple two-band decomposition with fixed gating can support strong frozen-probe performance. Finer decompositions are selected elsewhere: $V \in \{3, 4, 6\}$ on NCI1 and REDDIT-BINARY, and $V>2$ in eight of ten node-level selections (Table 2). Because the selected resolution varies across datasets and V is selected jointly with width, depth, and optimization settings, we keep V configurable.

Table 2: Transductive node-classification performance on five heterophilic benchmarks (Platonov et al., 2023). Values are mean ± population standard deviation across ten official splits after averaging five encoder seeds within each split. Accuracy is reported for roman-empire and amazon-ratings; ROC-AUC is reported for the remaining datasets, all on a 0–100 scale. V denotes the selected number of bands. The Δ row gives a descriptive best-of-two BMSAE comparison with the strongest displayed non-BMSAE result, including Raw+LR. Bold marks the highest mean per column, excluding Raw+LR. See Appendix B for experimental details.

Method	roman-emp. (acc, %)	amazon-rat. (acc, %)	minesw. (AUC, %)	tolokers (AUC, %)	questions (AUC, %)
Shallow references					
Raw+LR	63.7±0.6	38.3±0.4	51.1±1.0	71.9±0.8	69.8±1.1
SGC (Wu et al., 2019)	38.0±0.8	41.5±0.5	72.1±1.0	74.0±1.4	73.5±1.1
Unsupervised message passing reference					
GraphSAGE (Hamilton et al., 2017)	52.6±0.4	40.0±0.3	89.0±0.5	80.6±0.4	71.1±1.2
Contrastive self-supervised methods					
DGI (Veličković et al., 2019)	64.2±0.3	41.0±0.4	88.3±1.1	79.4±1.1	73.5±1.2
GRACE (Zhu et al., 2020)	56.0±0.5	40.4±0.3	87.6±0.9	78.6±1.0	72.4±1.0
GGD (Zheng et al., 2022)	57.0±0.5	39.0±0.7	88.8±1.0	77.4±1.3	73.4±1.2
CCA-SSG (Zhang et al., 2021)	59.4±0.5	40.3±0.3	86.3±0.9	76.9±0.8	73.9±0.9
BGRL (Thakoor et al., 2022)	62.6±0.3	40.1±0.3	83.0±1.0	77.0±1.0	72.1±1.0
GraphACL (Xiao et al., 2023)	65.1±0.5	41.3±0.3	85.1±0.9	79.8±1.1	73.4±1.1
Masked / generative self-supervised methods					
GraphMAE2 (Hou et al., 2023)	66.8±0.5	**41.6±0.3**	88.6±1.0	78.8±1.1	73.3±0.8
Spectral masked autoencoding					
BMSAE-Linear (ours)	**73.5±0.5** (V=3)	39.3±0.3 (V=2)	90.8±0.6 (V=4)	82.4±0.6 (V=8)	74.3±1.0 (V=6)
BMSAE-Sqrt (ours)	73.3±0.5 (V=2)	40.1±0.3 (V=8)	**90.9±0.6** (V=6)	**82.8±0.7** (V=4)	**74.7±1.0** (V=4)
Δ *best BMSAE − best baseline*	+6.7	−1.5	+1.9	+2.2	+0.8

6.3 Node-level representation learning

We focus the node-level evaluation on heterophilic graphs, a setting in which representations that retain relationships across multiple spectral bands are particularly relevant. BMSAE samples a band as a reconstruction target and predicts it from the remaining bands. The following experiments evaluate performance within this regime. We did not observe comparable gains on the homophilic benchmarks we examined.

Node-level results. Table 2 evaluates the two pre-specified BMSAE variants on five heterophilic graphs. The best BMSAE variant has the highest reported mean on four datasets: roman-empire (73.5, +6.7 over GraphMAE2), minesweeper (90.9, +1.9 over GraphSAGE), tolokers (82.8, +2.2 over GraphSAGE), and questions (74.7, +0.8 over CCA-SSG). Both variants exceed the strongest displayed baseline on each of those four datasets. GraphMAE2 remains stronger on amazon-ratings (41.6 versus the best BMSAE result of 40.1). Thus BMSAE attains the highest reported mean among the compared methods on four of the five evaluated heterophilic datasets. The two spacing variants select different V on every node dataset, yet their scores differ by at most 0.8 points, suggesting that performance is not tied to one particular band resolution. The largest gain occurs on roman-empire, where SGC, whose fixed propagation acts as a low-pass filter, scores 25.7 points below Raw+LR, whereas both BMSAE variants exceed Raw+LR by about 10 points. This pattern is consistent with spectral-band reconstruction being useful when relevant information spans multiple frequencies.

6.4 Ablation and robustness

Frozen spectral-feature control. To assess performance beyond fixed spectral preprocessing, we compare BMSAE with a linear probe on concatenated spectral bands, using the same selected filter banks and downstream evaluation procedures (procedure in Appendix C.1). Table 3 shows higher mean scores for BMSAE in all eight distinct graph configurations and all ten node configurations. The median improvement across the graph configurations is 2.8 percentage points. This includes the four V=2 graph configurations (PROTEINS +0.8, D&D +3.0, COLLAB +4.0, REDDIT-MULTI-5K +1.1), where the control is a linear probe on the same two fixed bands. Even these simple

Table 3: Comparison with frozen spectral features at the selected BMSAE configurations. BMSAE scores follow the reporting conventions of Tables 1 and 2. Frozen features are deterministic; their ± values are standard deviations across the graph-level CV10 folds or the ten official node splits. Node-level ± values are therefore directly comparable, whereas graph-level values summarize different sources of variation (encoder seeds for BMSAE, folds for frozen features). Δ_{frz} is BMSAE minus frozen features, computed from unrounded means. Graph-level $V=2$ configurations, which are identical under Linear and Sqrt spacing, are shown once.

Configuration	BMSAE	Frozen bands	Δ_{frz}	Metric
Graph classification				
PROTEINS ($V=2$, Linear/Sqrt)	**76.3**±0.3	75.5±3.9	+0.8	acc.
NCI1 ($V=6$, Linear)	**76.8**±0.2	72.9±1.5	+3.9	acc.
NCI1 ($V=4$, Sqrt)	**75.8**±0.8	72.1±1.7	+3.7	acc.
D&D ($V=2$, Linear/Sqrt)	**78.6**±0.6	75.6±4.1	+3.0	acc.
COLLAB ($V=2$, Linear/Sqrt)	**77.7**±0.4	73.8±2.1	+4.0	acc.
REDDIT-MULTI-5K ($V=2$, Linear/Sqrt)	**57.0**±0.2	55.9±1.5	+1.1	acc.
REDDIT-BINARY ($V=4$, Linear)	**93.1**±0.3	91.0±2.4	+2.1	acc.
REDDIT-BINARY ($V=3$, Sqrt)	**93.3**±0.1	90.6±2.0	+2.7	acc.
Node classification				
roman-empire ($V=3$, Linear)	**73.5**±0.5	71.4±0.5	+2.1	acc.
roman-empire ($V=2$, Sqrt)	**73.3**±0.5	71.7±0.7	+1.6	acc.
amazon-ratings ($V=2$, Linear)	**39.3**±0.3	38.4±0.4	+0.9	acc.
amazon-ratings ($V=8$, Sqrt)	**40.1**±0.3	38.8±0.3	+1.3	acc.
minesweeper ($V=4$, Linear)	**90.8**±0.6	89.4±0.4	+1.4	AUC
minesweeper ($V=6$, Sqrt)	**90.9**±0.6	90.1±0.6	+0.8	AUC
tolokers ($V=8$, Linear)	**82.4**±0.6	80.4±0.6	+2.0	AUC
tolokers ($V=4$, Sqrt)	**82.8**±0.7	79.9±0.8	+2.9	AUC
questions ($V=6$, Linear)	**74.3**±1.0	69.8±1.3	+4.5	AUC
questions ($V=4$, Sqrt)	**74.7**±1.0	71.2±1.2	+3.5	AUC

configurations still score higher than the frozen control, although the control does not separate the contributions of masked training, encoder architecture, and nonlinear transformation.

Robustness. On the graph-classification panel, performance changes little across the tested Chebyshev orders. Alternative encoder families also achieve competitive results on several datasets, although their relative performance varies. Appendix C reports the corresponding studies and runtime measurements.

7 Conclusion

BMSAE learns graph representations by predicting masked spectral bands from visible bands within a fixed Parseval frame. For ideal filters, Parseval synthesis ensures that masked-band prediction error controls feature-reconstruction error and fixed-predictor loss stability. Under the reported frozen-probe protocols, BMSAE achieves the highest displayed mean among methods other than Raw+LR on five of six graph-classification datasets and the highest mean among compared methods on four of five heterophilic node datasets, with weaker results on NCI1 and amazon-ratings. Together, the reconstruction guarantee and the graph- and node-level evaluations support band-masked spectral prediction as an effective pretext task for graph representation learning.

REFERENCES

Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V. Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowledge-Based Systems*, 256:109631, 2022. doi: 10.1016/j.knosys.2022.109631.

Deyu Bo, Yuan Fang, Yang Liu, and Chuan Shi. Graph contrastive learning with stable and scalable spectral encoding. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning (ICML)*, 2020.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. GraphMAE: Self-supervised masked graph autoencoders. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2022.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. GraphMAE2: A decoding-enhanced masked self-supervised graph learner. In *The Web Conference (WWW)*, 2023.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NeurIPS Workshop on Bayesian Deep Learning*, 2016.

Nora Leonardi and Dimitri Van De Ville. Tight wavelet frames on multislice graphs. *IEEE Transactions on Signal Processing*, 61(13):3357–3367, 2013.

Lu Lin, Jinghui Chen, and Hongning Wang. Spectral augmentation for self-supervised learning on graphs. In *International Conference on Learning Representations (ICLR)*, 2023.

Nian Liu, Xiao Wang, Deyu Bo, Chuan Shi, and Jian Pei. Revisiting graph contrastive learning from the perspective of graph spectrum. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022.

Ruyue Liu, Rong Yin, Yong Liu, and Weiping Wang. ASWT-SGNN: Adaptive spectral wavelet transform-based self-supervised graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13990–13998, 2024. doi: 10.1609/aaai.v38i12.29307.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. *ICML Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations (ICLR)*, 2023.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations (ICLR)*, 2020.

Qiaoyu Tan, Ninghao Liu, Xiao Huang, Soo-Hyun Choi, Li Li, Rui Chen, and Xia Hu. S2GAE: Self-supervised graph autoencoders are generalizable learners with graph masking. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 787–795, 2023. doi: 10.1145/3539597.3570404.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *International Conference on Learning Representations (ICLR)*, 2022.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations (ICLR)*, 2019.

Guancheng Wan, Yijun Tian, Wenke Huang, Nitesh V. Chawla, and Mang Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, 2024.

Felix Wu, Amauri H. Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *International Conference on Machine Learning (ICML)*, 2019.

Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S. Yu. From canonical correlation analysis to self-supervised graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. Spectral feature augmentation for graph contrastive learning and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11289–11297, 2023. doi: 10.1609/aaai.v37i9.26336.

Yizhen Zheng, Shirui Pan, Vincent C. S. Lee, Yu Zheng, and Philip S. Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *ICML Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020.

Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. An empirical study of graph contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

A Theoretical details and extensions

A.1 Proofs and additional consequences of the main theorem

Proof of Corollary 1. The Lipschitz assumption and the triangle inequality give

$$\left|\widehat{R}_{\hat{X}}(r) - \widehat{R}_X(r)\right| \leq \frac{L}{n} \sum_{v=1}^{n} \|\hat{x}_v - x_v\|_2 .$$

Cauchy–Schwarz bounds the right-hand side by $L \left\|\hat{X} - X\right\|_{\mathrm{F}} / \sqrt{n}$. The final inequality follows from Theorem 1 and the definition of $\mathcal{L}_M$.

A.2 Approximate Chebyshev reconstruction

The main result concerns the ideal Meyer responses. For the implemented order-K Chebyshev filters, write $\widetilde{T} : X \mapsto (\widetilde{\psi}_{j,K}(L_{\mathrm{sym}})X)_{j=1}^{V}$ and suppose

$$\left\|\widetilde{T}^*\widetilde{T} - I\right\|_{\mathrm{op}} \le \delta. \tag{4}$$

For approximate targets $\widetilde{Z}_j = \widetilde{\psi}_{j,K}(L_{\mathrm{sym}})X$, let Δ be zero on visible bands and satisfy $\Delta_j = \widehat{\widetilde{Z}}_j - \widetilde{Z}_j$ for $j \in M$.

Proposition 1 (Approximate-frame reconstruction bound)**.** *Under equation 4, approximate adjoint synthesis*

$$\widehat{X}_{\mathrm{approx}} = \widetilde{T}^*(\widetilde{T}X + \Delta)$$

obeys

$$\left\|\widehat{X}_{\mathrm{approx}} - X\right\|_{\mathrm{F}} \le \delta \|X\|_{\mathrm{F}} + \sqrt{1+\delta}\, \|\Delta\|_{\mathrm{F}}. \tag{5}$$

Proof. Decompose the error as

$$\widehat{X}_{\mathrm{approx}} - X = (\widetilde{T}^*\widetilde{T} - I)X + \widetilde{T}^*\Delta.$$

The first term is at most $\delta \|X\|_{\mathrm{F}}$. Moreover, $\left\|\widetilde{T}^*\right\|_{\mathrm{op}}^2 = \left\|\widetilde{T}^*\widetilde{T}\right\|_{\mathrm{op}} \le 1+\delta$, which bounds the second by $\sqrt{1+\delta}\,\|\Delta\|_{\mathrm{F}}$. The triangle inequality proves the result. □

This separates Chebyshev approximation from masked-band prediction error and reduces to the ideal result at $\delta = 0$. The numerical diagnostics in Appendix C are not certified bounds on equation 4; instantiating this proposition numerically requires a separately justified operator-norm bound.

A.3 Disjoint-union and graph-level extension

For disjoint graphs $\{G^{(g)}\}_{g=1}^{G}$ with feature matrices $X^{(g)}$, let $\mathcal{G} = \bigsqcup_g G^{(g)}$. We use $\rho_{X,g} : \mathbb{R}^{n_g \times p} \to \mathbb{R}^{d_X}$ for pooling reconstructed or original features; this differs from experimental latent pooling $\rho_H : \mathbb{R}^{n_g \times d_{\mathrm{lat}}} \to \mathbb{R}^{d_H}$.

Proposition 2 (Direct sums of Parseval frames are Parseval)**.** *If the bank is Parseval on every graph, then its bank on $\mathcal{G}$ is Parseval. For a shared mask M and $\varepsilon_g = \max_{j\in M} \left\|\hat{Z}_j^{(g)} - Z_j^{(g)}\right\|_{\mathrm{F}}$,*

$$\left\|\hat{X} - X\right\|_{\mathrm{F}} \le \sqrt{|M|}\Big(\sum_g \varepsilon_g^2\Big)^{1/2}. \tag{6}$$

Proof. Functional calculus respects block-diagonal structure, so $T = \bigoplus_g T^{(g)}$ and $T^*T = \bigoplus_g T^{(g)*}T^{(g)} = I$. The proof of Theorem 1 on the union gives $\|\Delta\|_{\mathrm{F}}^2 \le |M| \sum_g \varepsilon_g^2$, which yields equation 6. □

Corollary 2 (Graph prediction-loss stability under reconstruction)**.** *For graph labels $y_1, \dots, y_G$, fix a scoring rule r and define*

$$\widehat{R}_{\mathrm{orig}}(r) := \frac{1}{G}\sum_{g=1}^{G} \ell\Big(r\Big(\rho_{X,g}\Big(X^{(g)}\Big)\Big), y_g\Big),$$

with $\widehat{R}_{\mathrm{rec}}(r)$ replacing $X^{(g)}$ by $\hat{X}^{(g)}$. If $z \mapsto \ell(r(z), y)$ is L-Lipschitz uniformly in y and $\rho_{X,g}$ is $L_{\rho,g}$-Lipschitz, then

$$\begin{aligned}\left|\widehat{R}_{\mathrm{rec}}(r) - \widehat{R}_{\mathrm{orig}}(r)\right| &\le \frac{L}{G}\sum_{g=1}^{G} L_{\rho,g}\left\|\hat{X}^{(g)} - X^{(g)}\right\|_{\mathrm{F}} \\ &\le \frac{L}{G}\sum_{g=1}^{G} L_{\rho,g}\left(\sum_{j\in M}\left\|\hat{Z}_j^{(g)} - Z_j^{(g)}\right\|_{\mathrm{F}}^2\right)^{1/2}.\end{aligned} \tag{7}$$

Proof. Apply the Lipschitz conditions graph by graph, average, and then apply Theorem 1 separately to each graph. □

This is an empirical reconstructed-feature statement for fixed graphs and a fixed rule.

A.4 Pooling Lipschitz constants

Under Frobenius input and Euclidean output norms, mean pooling on n nodes has constant $1/\sqrt{n}$, sum pooling has constant $\sqrt{n}$, and coordinate-wise max pooling has constant at most 1. For unscaled concatenation, the constants combine in quadrature:

$$\mathrm{Lip}([\,\mathrm{mean}\,\|\,\mathrm{max}\,]) \le \sqrt{1+1/n}, \qquad \mathrm{Lip}([\,\mathrm{mean}\,\|\,\mathrm{sum}\,\|\,\mathrm{max}\,]) \le \sqrt{n+1+1/n}.$$

A.5 Non-tight frames and canonical-dual synthesis

The unit-constant result in Theorem 1 uses $T^*T = I$. For a frame with $\alpha I \preceq T^*T \preceq \beta I$, adjoint synthesis is not generally exact. Exact synthesis is recovered by

$$T^\dagger := (T^*T)^{-1}T^*, \qquad T^\dagger T = I.$$

With stacked-Frobenius coefficient and Frobenius feature norms, $\left\|T^\dagger\right\|_{\mathrm{op}} \le 1/\sqrt{\alpha}$. Thus non-tight frames retain exact stable reconstruction with their canonical dual. Parseval tightness permits exact synthesis using the adjoint itself, $T^* = T^\dagger$, with unit operator norm.

B Experimental protocol and hyperparameter search

B.1 Dataset descriptions and sources

Table 4 summarizes the datasets used in the two main comparison panels. The graph-classification panel uses standard TU benchmark graphs and their graph labels. The node-classification panel uses the five heterophilic single-graph benchmarks introduced by Platonov et al. (2023). We use the benchmark-provided node features when available; the featureless social TU datasets use the common five-dimensional Local Degree Profile (LDP) features (Cai & Wang, 2019).

Table 4: Datasets in the main comparison panels. Graph counts refer to the number of graphs in the TU collection, and node counts refer to the single observed node graph.

Panel	Dataset	Description and metric	Source / split
Graph	REDDIT-BINARY (2,000)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Graph	REDDIT-MULTI-5K (4,999)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Graph	COLLAB (5,000)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Graph	PROTEINS (1,113)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Graph	D&D (1,178)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Graph	NCI1 (4,110)	Graph classification, accuracy	TU (Morris et al., 2020); CV10
Node	roman-empire (22,662)	Node classification, accuracy	Platonov et al. (2023); 10 official splits
Node	amazon-ratings (24,492)	Node classification, accuracy	Platonov et al. (2023); 10 official splits
Node	minesweeper (10,000)	Node classification, ROC-AUC	Platonov et al. (2023); 10 official splits
Node	tolokers (11,758)	Node classification, ROC-AUC	Platonov et al. (2023); 10 official splits
Node	questions (48,921)	Node classification, ROC-AUC	Platonov et al. (2023); 10 official splits

B.2 Evaluation and BMSAE selection protocol

B.2.1 Shared BMSAE validation grid

Both graph- and node-classification experiments use the same candidate grid for each of the two pre-specified spacing variants:

$$V \in \{2,3,4,6,8\}, \qquad d_{\mathrm{lat}} \in \{128,256,512\},$$
$$\eta \in \{10^{-2},10^{-3}\}, \qquad L_{\mathrm{gate}} \in \{1,2\},$$
$$\lambda_{\mathrm{wd}} \in \{0,10^{-4},10^{-3},10^{-2}\}.$$

On the fixed interval $[0, 2.02]$, let $0 = t_0 < \cdots < t_{V-1} = 2.02$ be the V band centers, with center t_j defining response ψ_{j+1} (thus $j = 0, \ldots, V-1$ maps to the manuscript's band indices $1, \ldots, V$). For an interior center t_j, response ψ_{j+1} rises as $\sin\big(\frac{\pi}{2}(\mu - t_{j-1})/(t_j - t_{j-1})\big)$ on $[t_{j-1}, t_j]$ and falls as $\cos\big(\frac{\pi}{2}(\mu - t_j)/(t_{j+1} - t_j)\big)$ on $[t_j, t_{j+1}]$, and is zero elsewhere; ψ_1 and ψ_V retain only the in-interval falling and rising lobes, respectively. These responses are continuous and piecewise smooth, with derivative jumps where a lobe meets its zero extension. We use "Meyer" in the graph-spectral filter-bank sense: these are Meyer-style compactly supported raised-sine/cosine windows whose squared responses form a partition of unity on overlaps, rather than the classical infinitely differentiable Meyer scaling/wavelet pair. Adjacent squared responses sum to one on every overlap. BMSAE-Linear uses $t_j = 2.02j/(V-1)$, whereas BMSAE-Sqrt uses $t_j = 2.02(j/(V-1))^2$, which concentrates boundaries near lower frequencies. These fixed responses are not learned.

The main-comparison tables report these two rows, not a per-dataset maximum over spectral parameterizations. Every other method-level choice in the main comparisons is fixed for both variants: the gated per-band encoder (alternative BMSAE encoder families are evaluated in Appendix C), a separate linear decoder for every target band, one globally held-out whole band per self-supervised step, uniform target-band selection and reconstruction-loss weighting, Adam, one SSL step per epoch, a degree-K=30 Chebyshev approximation (with 31 coefficients), and 400 SSL epochs. The held-out band is excluded from the encoder input for every node in that update; it is not replaced by zeros. The objective is the full-node reconstruction loss, and every update uses all nodes except on REDDIT-MULTI-5K (about 2.5M nodes). There, each epoch's single update draws 32,768 nodes uniformly without replacement and averages the masked-band squared error over the sampled nodes, feature dimensions, and masked bands. Because the Gate encoder and decoders act row-wise, this sampled loss and its gradient are unbiased estimates of the full-node loss for the sampled mask. Frozen embeddings are still computed for all nodes, in 65,536-node no-gradient chunks. The gate uses independent per-band maps and a dataset-level softmax gate over the visible bands. Gate depth one is $\mathrm{Linear}(p, d_{\mathrm{lat}}) \to \mathrm{GELU}$ per band; depth two adds $\mathrm{Linear}(d_{\mathrm{lat}}, d_{\mathrm{lat}}) \to \mathrm{GELU}$. No message-passing layer or band concatenation is added by either choice. The Linear and Sqrt variants share the V=2 bank, so those 48 $(d_{\mathrm{lat}}, \eta, \lambda_{\mathrm{wd}}, L_{\mathrm{gate}})$ cells are trained once and are eligible for either variant.

B.2.2 Graph-level classification

For Table 1, each unsupervised encoder is trained label-free on all graphs and then evaluated with a frozen downstream linear probe. Thus SSL may use every graph's features and structure, including graphs assigned to a particular probe fold, but it never accesses graph labels.

The TU benchmarks use one fixed stratified 10-fold partition per dataset (split seed 0). For each fold, the C=1 probe is fit on the labels of the other nine folds and scored on that fold's labels; its CV10 mean averages the ten held-out-fold scores. All deep methods use identical node features, the mean‖sum‖max read-out, frozen L_2-regularized linear probe (C=1), and fixed folds; only the self-supervised encoder differs.

For each candidate configuration, we train one seed-0 encoder and select the configuration with the highest fixed-CV10 mean, separately for each dataset and BMSAE variant. Configuration selection and reporting therefore use the same fixed folds; no separate validation split is held out. No labels enter encoder training or probe-regularization tuning.

After configuration selection, each learned encoder is evaluated with seeds $\{0, 1, 2, 3, 4\}$. For each seed, we average probe accuracy over the fixed ten folds; Table 1 reports the mean and population standard deviation of these five CV10 averages. Raw+LR and WL are deterministic references and report a single CV10 mean; WL uses an SVM on its precomputed kernel under the same folds.

B.2.3 Node-level classification

For Table 2, all methods follow the same transductive protocol. First, the self-supervised encoder trains without labels on the complete observed graph $(A_{\mathrm{full}}, X_{\mathrm{full}})$. Second, the frozen encoder embeds all nodes of that graph. Third, a GPU-LBFGS ℓ_2-regularized linear probe (C=1) is fit only on training-node latents and evaluated on validation and test nodes of the same graph. Every dataset uses the 10 predefined 50/25/25 Platonov splits.

Table 5: Gate-encoder BMSAE configurations selected for the two main comparison panels. Graph rows are selected by the fixed-CV10 procedure; node rows are selected using validation performance only, without test-score selection. d_{lat} is latent width, η is learning rate, λ_{wd} is weight decay, and L_{gate} is gate depth. Linear and Sqrt rows with $V{=}2$ share the same physical two-band bank.

Panel	Dataset / variant	V	d_{lat}	η	λ_{wd}	L_{gate}
Graph	REDDIT-BINARY, Linear	4	512	10^{-2}	0	1
	REDDIT-BINARY, Sqrt	3	512	10^{-2}	10^{-4}	1
	REDDIT-MULTI-5K, Linear	2	512	10^{-2}	0	1
	REDDIT-MULTI-5K, Sqrt	2	512	10^{-2}	0	1
	COLLAB, Linear	2	128	10^{-2}	0	2
	COLLAB, Sqrt	2	128	10^{-2}	0	2
	PROTEINS, Linear	2	256	10^{-2}	10^{-2}	1
	PROTEINS, Sqrt	2	256	10^{-2}	10^{-2}	1
	D&D, Linear	2	128	10^{-2}	10^{-3}	1
	D&D, Sqrt	2	128	10^{-2}	10^{-3}	1
	NCI1, Linear	6	512	10^{-2}	0	2
	NCI1, Sqrt	4	256	10^{-2}	0	1
Node	roman-empire, Linear	3	512	10^{-2}	10^{-4}	1
	roman-empire, Sqrt	2	512	10^{-2}	0	1
	amazon-ratings, Linear	2	512	10^{-3}	10^{-2}	1
	amazon-ratings, Sqrt	8	256	10^{-3}	10^{-3}	1
	minesweeper, Linear	4	512	10^{-2}	0	1
	minesweeper, Sqrt	6	512	10^{-3}	0	2
	tolokers, Linear	8	512	10^{-3}	0	1
	tolokers, Sqrt	4	512	10^{-3}	0	2
	questions, Linear	6	256	10^{-3}	10^{-3}	1
	questions, Sqrt	4	512	10^{-2}	0	2

For each dataset and spacing variant, each candidate is screened with encoder seed 0 and selected by its mean validation score across the ten official splits; test scores do not enter selection. The selected configuration is retrained with encoder seeds $\{0, 1, 2, 3, 4\}$, and test scores are averaged across seeds within each split. Table 2 reports the mean and population standard deviation across the resulting ten split-level averages. These bars therefore mainly reflect split variation and are not directly comparable to the graph-level seed-only bars. The released main-comparison Gate sweep records every dataset, spacing, V, latent width, learning rate, weight decay, Gate depth, optimizer, masking regime, Chebyshev order, encoder seed, split index, validation score, and post-selection test score.

B.3 Baseline provenance and hyperparameter search

Learned baselines are selected and evaluated with the same input features and the same graph- and node-level procedures as BMSAE (Appendices B.2.2 and B.2.3), with each method's own search space in place of the BMSAE grid. Below, we document implementation sources, departures from the released code, and hyperparameter search spaces.

Implementation sources. Every baseline runs either the authors' official open-source release or the canonical implementation shipped with PyTorch Geometric (PyG) (Fey & Lenssen, 2019), as listed in Table 6. The only encoder we implement ourselves is SGC, which we implement directly in PyTorch following Equation 11 of Wu et al. (2019).

Hyperparameter search. For each learned method, searches start from reference-paper or released-code settings and vary shared capacity and optimization choices where applicable: learning rate $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, hidden dimension $\{128, 256, 512\}$, and depth $\{2, 3, 4\}$. Each method additionally sweeps its relevant objective controls in a small reference-informed range, such as augmentation or mask rate, dropout, contrastive temperature, or EMA decay. This gives every method an opportunity to tune the choices that define its objective without imposing BMSAE-specific hyperparameters on other methods.

Table 6: References and implementation sources for the comparison rows. The search and evaluation protocol is described in the surrounding text.

Method	Reference	Code source
Graph classification (Table 1)		
Raw+LR	—	Shared evaluation implementation
WL	Shervashidze et al. (2011)	GraKeL + scikit-learn
InfoGraph	Sun et al. (2020)	PyGCL (Zhu et al., 2021)
GraphCL	You et al. (2020)	PyGCL (Zhu et al., 2021)
GraphMAE	Hou et al. (2022)	Official release
S2GAE	Tan et al. (2023)	Official release
Sp2GCL	Bo et al. (2023)	Official release
Node classification (Table 2)		
Raw+LR	—	Shared evaluation implementation
SGC	Wu et al. (2019)	Direct implementation
GraphSAGE	Hamilton et al. (2017)	PyG (Fey & Lenssen, 2019)
DGI	Veličković et al. (2019)	PyG (Fey & Lenssen, 2019)
GRACE	Zhu et al. (2020)	Official release
GGD	Zheng et al. (2022)	Official release
CCA-SSG	Zhang et al. (2021)	Official release
BGRL	Thakoor et al. (2022)	Official release
GraphACL	Xiao et al. (2023)	Official release
GraphMAE2	Hou et al. (2023)	Official release

WL kernel. Unlike the learned methods, WL selects its subtree depth and SVM regularization within each outer-training fold.

C Robustness, ablations, and runtime

C.1 Frozen spectral-feature control

Table 3 compares each selected BMSAE configuration with a matched fixed-feature control. Graph configurations follow the Table 1 fixed-CV10 selection procedure; node configurations are validation-selected. *Frozen spectral features* concatenate all fixed Meyer coefficient bands of the selected bank and apply the same $C{=}1$ linear probe, without learning an encoder. BMSAE is evaluated over five encoder seeds. Frozen features are deterministic and therefore have one probe evaluation per fixed CV10 split (graph) or official split (node).

C.2 Numerical approximation diagnostics

The exact Parseval identities concern ideal Meyer responses; finite-order Chebyshev filters need not be exactly tight. The analytic responses, Chebyshev coefficient generation, and the uniform 200,001-point grid evaluation in this subsection use NumPy `float64`. On that grid over $[0, 2.02]$, we report

$$\delta_{K,\mathrm{grid}} := \max_{\mu \text{ on the grid}} \left| \sum_j \widetilde{\psi}_{j,K}(\mu)^2 - 1 \right|.$$

For the nine unique fixed-$[0, 2.02]$ Meyer banks evaluated here, ideal filters have observed maximum grid defect $2.22{\times}10^{-16}$. At $K{=}30$, the largest approximate-filter grid defect is $5.28{\times}10^{-2}$ for Sqrt-$V{=}8$. These `float64` grid diagnostics are not the numerical precision of the reported filtering path: the batched GPU recurrence, sparse matrix values, input features, and returned band embeddings use `float32`. The grid figures are numerical diagnostics, not certified continuous-domain or operator-norm bounds; see Appendix A.2 for the conditional formal result.

Table 7: `float64` dense-grid order-30 approximation diagnostics for every unique fixed-$[0, 2.02]$ Meyer bank evaluated here. ϵ_∞ is the observed maximum filter-response error, $\delta_{30,\mathrm{grid}}$ is the observed maximum grid defect, and κ_{grid} is the grid-derived ratio $(1 + \delta_{30,\mathrm{grid}})/(1 - \delta_{30,\mathrm{grid}})$; these quantify the analytic/grid calculation rather than runtime `float32` filtering, and neither is a certified continuous-domain bound.

Bank	ϵ_∞	$\delta_{30,\mathrm{grid}}$	κ_{grid}
Linear-V=2	8.66e−15	1.73e−14	1.00000
Linear-V=3	1.52e−2	3.74e−3	1.00750
Linear-V=4	3.20e−2	2.23e−3	1.00448
Linear-V=6	4.48e−2	2.34e−2	1.04800
Linear-V=8	8.80e−2	2.41e−2	1.04929
Sqrt-V=3	2.16e−2	6.76e−3	1.01360
Sqrt-V=4	4.53e−2	2.14e−2	1.04374
Sqrt-V=6	1.28e−1	1.15e−2	1.02318
Sqrt-V=8	1.68e−1	5.28e−2	1.11147

Table 8: Seed-paired fixed-CV10 differences in accuracy points relative to K=30: mean ± population standard deviation of $\Delta_{K-30,s} = \mathrm{CV10}_{K,s} - \mathrm{CV10}_{30,s}$ across the same five encoder seeds. Positive values favour the alternate order. Linear/Sqrt rows are shared physical V=2 banks. These descriptive summaries are not statistical-equivalence tests.

Configuration	K=5	K=10	K=20	K=50
PROTEINS (Linear/Sqrt)	−0.04±0.07	+0.05±0.07	+0.00±0.00	+0.00±0.00
NCI1 (Linear)	−0.07±0.75	−0.46±0.61	−0.09±0.27	−0.15±0.55
NCI1 (Sqrt)	+0.44±0.84	+0.00±0.72	−0.03±0.25	+0.07±0.29
D&D (Linear/Sqrt)	+0.10±0.16	−0.02±0.03	+0.00±0.00	+0.00±0.00
COLLAB (Linear/Sqrt)	−0.02±0.33	−0.21±0.21	+0.01±0.20	−0.03±0.20
REDDIT-MULTI-5K (Linear/Sqrt)	+0.11±0.30	+0.06±0.26	+0.16±0.20	+0.20±0.20
REDDIT-BINARY (Linear)	−0.04±0.23	−0.15±0.31	−0.07±0.15	−0.36±0.17
REDDIT-BINARY (Sqrt)	−0.20±0.04	+0.03±0.27	+0.17±0.09	+0.11±0.16

C.3 Robustness to Chebyshev order

The spectral bands are approximated by a Chebyshev expansion of order K (fixed to 30 in all main results). To test downstream sensitivity without introducing a second selection step, we hold every Table 1 selected fixed-interval configuration fixed and change only $K \in \{5, 10, 20, 30, 50\}$. The completed K=30 bars are reused; each alternate order is retrained at encoder seeds 0–4 with the same fixed CV10 split, frozen C=1 probe, architecture, masking, optimizer, learning rate, weight decay, and batching mode. All configurations except REDDIT-MULTI-5K use one full-node reconstruction update per epoch. On REDDIT-MULTI-5K, each epoch reconstructs an independently drawn uniform subset of 32,768 nodes, and frozen full-node embeddings are encoded in 65,536-node no-gradient chunks for the probe. We compare every alternate order with K=30 by seed-paired CV10 differences: for seed s, $\Delta_{K-30,s} = \mathrm{CV10}_{K,s} - \mathrm{CV10}_{30,s}$. Table 8 reports the five-seed mean and population standard deviation of these differences. The shared V=2 Linear/Sqrt bank is trained once and mapped to both variants. This is an appendix robustness analysis, not a new selection rule. Figure 2 instead displays absolute fixed-CV10 scores (mean ± population standard deviation across the five seeds), rather than paired-difference standard deviations. Across all fixed selected configurations, the largest absolute paired mean change is 0.46 percentage points (NCI1 Linear at K=10). These descriptive paired results are not a statistical-equivalence test. Thus the Table 1 results are stable across the tested orders; K=30 remains a fixed implementation choice rather than a selected order. Numerical approximation diagnostics are reported above in Table 7.

C.4 Robustness to encoder architecture

Table 9 compares the Table 1 BMSAE-Sqrt Gate reference against BMSAE instances that replace the gated per-band encoder with independently selected MLP or Attention encoder families, holding the objective, masking, Chebyshev band construction, and probe fixed. For each alternative family and dataset, the fixed-CV10 sweep-and-retrain evaluation protocol, including splits, probe, encoder-

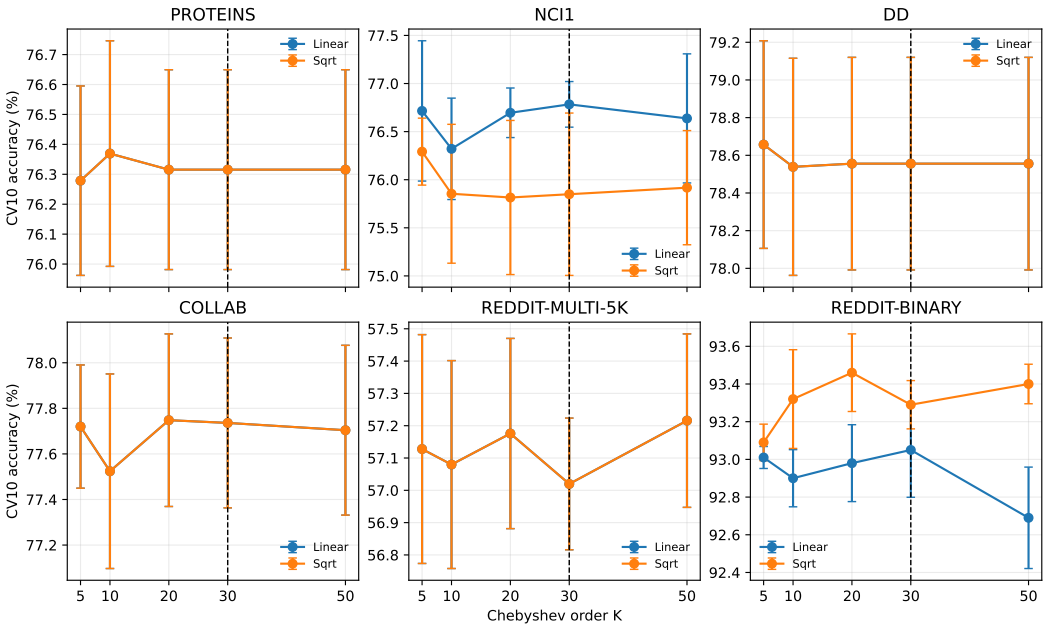

Figure 2: Five-seed fixed-CV10 graph-classification accuracy as the Chebyshev order varies under the adopted Table 1 Gate selections, with each Linear/Sqrt curve holding its selected configuration fixed and changing only K. Points are absolute fixed-CV10 accuracy means $\pm$ population standard deviations over the same five encoder seeds; they do not show paired-difference standard deviations. The dashed line marks the reported $K{=}30$. The Linear and Sqrt curves coincide where their selected $V{=}2$ bank is shared.

Table 9: Five-seed graph-classification BMSAE encoder-family ablation under the Sqrt protocol (accuracy, %). Entries are mean $\pm$ population standard deviation, rounded to one decimal place. Gate is the adopted Table 1 BMSAE-Sqrt reference; all three columns instantiate BMSAE with the same objective, uniform single-band masking, Chebyshev bands, and probe, differing in the per-band encoder family, with each family's hyperparameters independently selected under the fixed-CV10 protocol. Bold marks the best mean within each dataset before rounding.

Dataset	Gate	MLP	Attention
PROTEINS	**76.3**$\pm$0.3	75.0$\pm$0.8	74.2$\pm$1.1
NCI1	75.8$\pm$0.8	**75.9**$\pm$0.6	74.4$\pm$0.5
D&D	78.6$\pm$0.6	76.4$\pm$0.1	**79.0**$\pm$0.4
COLLAB	**77.7**$\pm$0.4	77.1$\pm$0.5	77.0$\pm$0.3
REDDIT-MULTI-5K	**57.0**$\pm$0.2	56.9$\pm$0.3	56.3$\pm$0.3
REDDIT-BINARY	**93.3**$\pm$0.1	92.2$\pm$0.5	93.1$\pm$0.2

seed aggregation, and shared search grid, is held fixed; the encoder family and all architecture-specific configuration choices are independently selected within that grid. Gate uses the adopted Table 1 BMSAE-Sqrt reference. Several means are close: Gate and MLP on NCI1 (75.8 versus 75.9) and REDDIT-MULTI-5K (57.0 versus 56.9), and Gate and Attention on D&D (78.6 versus 79.0) and REDDIT-BINARY (93.3 versus 93.1). The remaining comparisons differ by 0.6–2.2 points. Attention is best on D&D, MLP is best on NCI1, and Gate is best on the other four datasets.

C.5 Measured runtime and filtering cost

Table 10 reports end-to-end wall-clock per training seed for the main-comparison Gate BMSAE timing configurations and every deep baseline under the shared protocol, measured on a single 22 GB A10G GPU. For every method, timing starts at its fresh dataset-loading call and ends after SSL

training, frozen embedding extraction, and frozen $C=1$ probe evaluation. Thus method-specific input feature construction and preprocessing performed by that run are included. BMSAE measurements use the batched GPU implementation with the fixed spectral interval $[0, 2.02]$.

Table 10: Wall-clock seconds, mean ± population standard deviation over five solo encoder seeds, with every run alone on its GPU. R-BINARY and R-MULTI-5K abbreviate REDDIT-BINARY and REDDIT-MULTI-5K. Baseline times run from fresh dataset loading through SSL training, frozen embedding extraction, and probe evaluation. Each BMSAE total combines a separately measured fresh dataset load, feature construction, and band-construction pass with that seed's SSL training, frozen embedding extraction, and probe evaluation; "Band setup" is the former and "Chebyshev only" is the filtering portion within it, so neither is an additional cost. BMSAE timing data come from the fixed-$[0, 2.02]$ Table 1 main-comparison Gate re-selection runs. BMSAE times use the Table 5 configurations (Sqrt on R-BINARY, Linear on NCI1, and the shared $V=2$ bank elsewhere); baselines use their Table 1 selections.

Method	R-BINARY	R-MULTI-5K	COLLAB	PROTEINS	D&D	NCI1
BMSAE total	101±0	**101±0**	**31±0**	**8±0**	**15±0**	60±0
Band setup	28	75	21	4	5	8
Chebyshev only	2.2	5.0	5.2	1.1	2.3	3.6
InfoGraph	53±7	142±2	101±47	10±0	29±5	**26±1**
GraphCL	**43±1**	122±8	107±14	14±3	37±6	36±4
GraphMAE	266±5	734±67	54±8	57±5	168±55	958±265
S2GAE	544±3	1577±3	3812±17	46±0	743±3	65±2
Sp2GCL	1593±96	6007±574	595±222	101±23	748±238	220±10

BMSAE accesses the spectrum through degree-K Chebyshev polynomial filters of the normalized Laplacian, evaluated using sparse adjacency operations. In its batched GPU path, the V filters share a single sparse Chebyshev-basis recurrence: the sparse basis construction costs $O\big(K(|E| + n)p\big)$ and the V dense per-band accumulations cost $O(VKnp)$, without an eigendecomposition or a dense $n \times n$ matrix. The "Chebyshev only" row isolates this filtering step: it is 1.1–5.2 s across these six datasets. The larger "Band setup" row additionally includes feature construction, graph loading, CPU sparse assembly, and transfers; it should not be interpreted as filter time alone.

The direct measurements establish runtime and filtering cost only for these six tested datasets and timing configurations. End-to-end time also depends on the selected latent width, band count, and main-comparison Gate encoder, so sparse filtering alone does not imply that BMSAE is the fastest method in every column. Within the reported BMSAE runs, the measured Chebyshev-only times show only that Chebyshev filtering is not the dominant cost: the broader band-setup stage accounts for about 74% of the REDDIT-MULTI-5K total and 68% of the COLLAB total. BMSAE is fastest on REDDIT-MULTI-5K, COLLAB, PROTEINS, and D&D under the reported main-comparison Gate configurations.

AI ASSISTANCE DISCLOSURE

AI tools were used for writing polish and for retrieval and discovery, including locating potentially relevant related work and documentation. The authors reviewed and verified all citations, technical claims, experimental procedures, numerical results, and final manuscript text, and retain full responsibility for the paper's content.