BERNNET REVISITED: ADAPTIVE SPECTRAL WARPING AND DECOMPOSED REGULARISATION FOR HETEROPHILIC GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Spectral graph neural networks on heterophilic graphs face two structural limitations: fixed polynomial bandwidth, which allocates capacity to uninformative frequency regions, and homogeneous regularization, which cannot simultaneously capture smooth trends and sharp spectral transitions. This issue is further compounded by self-information dilution under high-order neighborhood aggregation. We present **BernNetII**, a framework that resolves each limitation through principled modifications to the Bernstein spectral filter: a learnable spectral warp that adaptively concentrates polynomial resolution on the graph's effective frequency band; and a morphological decomposition of filter coefficients into smooth and step components with individually tailored regularisation, augmented by a residual bypass that preserves node-level features under high-order propagation. We establish a Rademacher complexity bound with separate capacity control for each component and prove convergence of the joint proximal gradient optimisation. Experiments on eight node classification benchmarks demonstrate that BernNetII consistently outperforms or matches state-of-the-art methods, validating its effectiveness across diverse settings. Our code is available at https://github.com/AI-FZY/BernNetII.

1 INTRODUCTION

Graph neural networks (GNNs) have achieved strong performance across a wide range of applications, including social influence prediction (Qiu et al., 2018) and recommendation systems (Wu et al., 2019). Spectral GNNs formulate convolution in the Laplacian frequency domain, providing a principled framework for learning graph filters (Shuman et al., 2013). Beginning with ChebNet (Defferrard et al., 2016), which introduced truncated Chebyshev expansions as an efficient polynomial parameterization, subsequent work has shown that alternative bases, including the monomial basis (Chien et al., 2020), the Bernstein basis (He et al., 2021), and Chebyshev interpolation (He et al., 2022), can approximate a broad class of spectral filters and achieve strong empirical performance on homophilic benchmarks. However, this picture changes on heterophilic graphs, where connected nodes are likely to belong to different classes (Zhu et al., 2020): an effective filter must simultaneously preserve smooth global structure and sharply localise the narrow high-frequency bands that carry discriminative signals. Satisfying both requirements simultaneously with a single polynomial basis is the core challenge we address.

An ideal heterophilic filter decomposes into a smooth low-frequency component and a sharp step component, each with a distinct statistical structure. The smooth part is naturally regularised by an ℓ_2-type curvature penalty, whereas the step component calls for an ℓ_1-type sparsity constraint. Existing approaches apply a single regularizer to a unified coefficient vector $\boldsymbol{\theta}$, which induces an inherent trade-off and degrades both components (Section 2.4). This single-vector parameterization also makes the filter prone to Gibbs-type oscillations when approximating sharp spectral transitions (Hammond et al., 2011). The issue is further aggravated by high-order polynomial propagation, where repeated neighborhood aggregation progressively dilutes each node's own features. This effect is particularly severe in heterophilic settings, as the filter cannot effectively suppress conflicting signals from cross-class neighbors. Figure 1 illustrates these failure modes.

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.

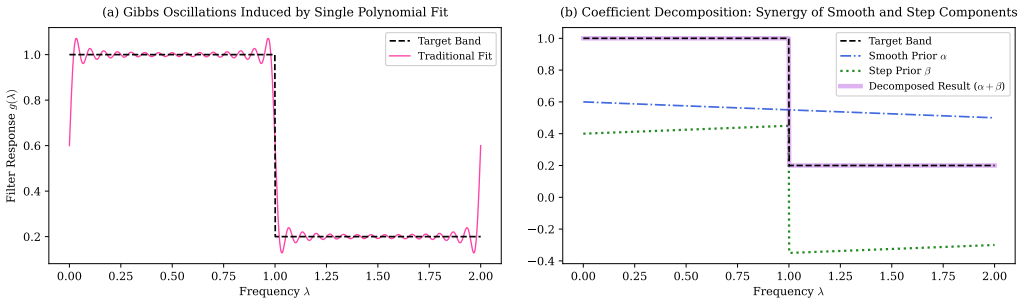

Figure 1: Left: a standard Bernstein filter (pink) approximating a band-pass target (black dashed) produces Gibbs-type oscillations outside the target band. Right: decomposing the filter coefficients into a smooth component $\boldsymbol{\alpha}$ and a step component $\boldsymbol{\beta}$ with individually tailored regularisation yields a sum that recovers the target accurately without spurious oscillations

To address this issue, we adopt the Bernstein basis, whose coefficients provide a direct and interpretable correspondence to the filter shape (Section 2.3), a property not available in the Chebyshev basis. We then decompose the filter coefficients as $\boldsymbol{\theta} := \boldsymbol{\alpha} + \boldsymbol{\beta}$. A second-order curvature penalty (Tibshirani et al., 2005) is applied to $\boldsymbol{\alpha}$, while a fused lasso penalty is imposed on $\boldsymbol{\beta}$, with the two components coupled through a position-adaptive curvature alignment term. We show that this decomposition strictly dominates any single-vector parameterization when the target filter contains a sufficiently sharp transition (Proposition 3). A learnable residual bypass τX is incorporated into the forward pass to mitigate feature dilution, without introducing additional inference parameters, since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are combined into $\boldsymbol{\theta}$ after training.

However, resolving the regularisation conflict alone does not address a second structural problem: both the Chebyshev and Bernstein bases distribute their $K+1$ degrees of freedom uniformly over the full spectrum $[0, 2]$, allocating substantial polynomial capacity to uninformative frequency regions. In heterophilic graphs, where discriminative signals concentrate in a narrow band, this fixed resolution wastes model capacity regardless of the basis or regularisation scheme used. Increasing K does not help: it scales both useful and wasted capacity by the same factor, leaving the inefficiency structurally unchanged (Section 2.5). We introduce a learnable spectral warp $s \in (0, 1)$ that rescales the Bernstein evaluation domain from $[0, 1]$ to $[0, s]$. This transformation restricts the realised response of the basis functions to an effective frequency sub-band governed by s, without requiring additional sparse matrix multiplications. We further show that the discrete total variation bound is preserved independently of s (Proposition 4), so the improvement in approximation efficiency does not compromise regularization control. The warp parameter is learned end-to-end via gradient descent, eliminating the need for manual bandwidth selection.

The resulting model, **BernNetII**, inherits the Pascal triangle recurrence of BernNet (He et al., 2021) and adds a spectral warp, a coefficient decomposition, and a residual bypass, with a Rademacher complexity bound (Proposition 6) and a convergence guarantee for the joint proximal gradient optimisation (Theorem 5).

2 Preliminaries

The decomposition and warp mechanisms introduced in Section 3 depend critically on the choice of polynomial basis: we show that the Chebyshev basis is structurally incompatible with coefficient-level decomposition, then identify three properties of the Bernstein basis that make it a suitable foundation, and formalise the two structural limitations that motivate our design.

2.1 Notation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $N = |\mathcal{V}|$ nodes, adjacency matrix $A \in \{0, 1\}^{N \times N}$, and degree matrix $D = \mathrm{diag}(d_1, \ldots, d_N)$ where $d_i = \sum_j A_{ij}$. The symmetric normalised Laplacian

$\tilde{L} = I - D^{-1/2}AD^{-1/2}$ admits the eigendecomposition

$$\tilde{L} = U\Lambda U^\top, \text{with} \quad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N), \text{and} \quad 0 \leq \lambda_1 \leq \cdots \leq \lambda_N \leq 2. \tag{1}$$

The homophily ratio

$$\mathcal{H}(\mathcal{G}) = \frac{|\{(u,v) : (u,v) \in \mathcal{E} \wedge y_u = y_v\}|}{|\mathcal{E}|}, \tag{2}$$

measures the fraction of edges that connect same-class nodes (Zhu et al., 2020). A graph spectral filter $g : [0, 2] \to \mathbb{R}$ acts on node features $X \in \mathbb{R}^{N\times d}$ via $g(\tilde{L})X = U\,\text{diag}(g(\lambda_1), \ldots, g(\lambda_N))U^\top X$, which would require $\mathcal{O}(N^3)$ eigendecomposition if computed directly. Polynomial approximation with a basis $\{\phi_k\}_{k=0}^K$ gives $p(\tilde{L})X = \sum_{k=0}^K \theta_k \phi_k(\tilde{L})X$, computable by matrix-vector products alone. The Chebyshev basis (Defferrard et al., 2016; He et al., 2022; Mason & Handscomb, 2002) is $T_k(\hat{\lambda}) = \cos(k \arccos \hat{\lambda})$ with $\hat{\lambda} = 2\lambda/\lambda_{\max} - 1 \in [-1, 1]$, and the Bernstein basis (He et al., 2021; Farouki, 2012) is $B_{k,K}(x) = \binom{K}{k} x^k (1-x)^{K-k}$ with $x = \lambda/2 \in [0, 1]$. The discrete total variation (DTV) of a coefficient sequence is $\|\boldsymbol{\theta}\|_{DTV} := \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k|$.

2.2 Structural Limitations of the Chebyshev Basis

The Chebyshev basis provides near-minimax approximation for smooth functions (He et al., 2022; Mason & Handscomb, 2002). However, Chebyshev-based GNNs consistently underperform Bernstein-based methods on heterophilic benchmarks (He et al., 2021; Defferrard et al., 2016). We argue that this gap arises from a structural limitation rather than from suboptimal training.

Each basis function T_k oscillates over the entire domain $[-1, 1]$, with frequency increasing in k (see Figure 4 in (He et al., 2022)). As a result, each coefficient $c_k = \frac{2}{K+1}\sum_j g(x_j)T_k(x_j)$ aggregates contributions from all frequency components. Consequently, imposing structured regularization on c_k, for example $\sum_k |c_{k+1} - c_k|$, does not induce a corresponding constraint on the shape of $g(\lambda)$ in the spectral domain, as there is no monotonic relationship between the two.

ChebNetII (He et al., 2022) attributes part of the performance degradation of ChebNet to coefficients that violate the analytic decay condition $|c_k| = \mathcal{O}(k^{-q})$, and shows that enforcing this constraint improves empirical results. However, even with this modification, the lack of transparency between Chebyshev coefficients and the frequency response remains. In particular, it is not possible to independently regularize the smooth and step components of a filter through Chebyshev coefficients alone. Furthermore, the interpolation nodes of the Chebyshev basis are fixed over $[-1, 1]$ and cannot adapt to the effective frequency range of a given graph.

These properties make the Chebyshev basis incompatible with the coefficient-level decomposition required by our framework, and motivate the use of the Bernstein basis.

2.3 The Bernstein Basis as a Structural Foundation

The Bernstein basis exhibits three properties that are central to our framework and are not shared by the Chebyshev basis. Together, they establish a direct and interpretable correspondence between coefficient constraints and filter shape, precisely the foundation that the decomposition in Section 3.1 requires.

Convex hull property. The partition of unity $\sum_k B_{k,K}(x) \equiv 1$ together with non-negativity $B_{k,K}(x) \geq 0$ implies that $g(\lambda) = \sum_k \theta_k B_{k,K}(\lambda/2)$ lies within the convex hull of its coefficients: $g(\lambda) \in [\min_k \theta_k, \max_k \theta_k]$ for all $\lambda \in [0, 2]$. Each coefficient θ_k therefore anchors the filter response near frequency $\lambda \approx 2k/K$, so any constraint on $\{\theta_k\}$ translates directly into an interpretable change in filter shape.

Variation-diminishing property. The Bernstein operator satisfies the variation-diminishing property (Lorentz, 2012): the number of sign changes of $g(\lambda)$ on $[0, 2]$ does not exceed that of the coefficient sequence $(\theta_0, \ldots, \theta_K)$. A DTV penalty on $\boldsymbol{\theta}$ therefore directly controls the oscillatory complexity of g.

TV bridge. These two properties together yield the following result, which is the mathematical foundation of our regularisation design.

Theorem 1 (DTV upper bound). *Let g be parameterised by Bernstein coefficients $\boldsymbol{\theta} \in \mathbb{R}^{K+1}$ as above. Then $\|g\|_{TV} \leq \|\boldsymbol{\theta}\|_{DTV}$, where $\|g\|_{TV} := \int_0^2 |g'(\lambda)| d\lambda$.*

Proof. The Bernstein derivative formula $B'_{k,K}(x) = K(B_{k-1,K-1}(x) - B_{k,K-1}(x))$ and the chain rule give $g'(\lambda) = \frac{K}{2} \sum_{k=0}^{K-1} (\theta_{k+1} - \theta_k) B_{k,K-1}(\lambda/2)$. Integrating $|g'|$ over $[0, 2]$, applying the triangle inequality, and using $\int_0^1 B_{k,K-1}(u) du = 1/K$ yields the bound. □

This correspondence does not hold for Chebyshev coefficients, indicating that the required structural property is specific to the Bernstein parameterization. We next show that, despite these advantages, existing Bernstein-based methods exhibit two structural limitations. The first motivates the coefficient decomposition introduced in Section 3.1.

2.4 Regularisation Conflict and Feature Dilution

The ideal filter for a heterophilic graph decomposes as $g^* = g^{\text{smooth}} + g^{\text{step}}$, inducing a corresponding split $\boldsymbol{\theta}^* = \boldsymbol{\theta}^s + \boldsymbol{\theta}^j$, where $\boldsymbol{\theta}^s$ varies slowly and $\boldsymbol{\theta}^j$ contains localised large differences. The smooth component is well captured by a second-order curvature penalty

$$\mu \|\Delta^2 \boldsymbol{\theta}\|_2^2 = \mu \sum_{k=0}^{K-2} (\theta_{k+2} - 2\theta_{k+1} + \theta_k)^2, \tag{3}$$

which promotes affine-like sequences, while the step component calls for a DTV penalty $\gamma \|\boldsymbol{\theta}\|_{DTV}$, which encourages piecewise-constant structures with few transitions. Applied simultaneously to a single vector $\boldsymbol{\theta}$, however, these two penalties interfere: the curvature penalty smooths out the jumps in $\boldsymbol{\theta}^j$ and weakens spectral cutoffs, while the DTV penalty drives $\boldsymbol{\theta}^s$ toward a step-like form and destroys low-frequency fidelity. The result is a trade-off that satisfies neither requirement.

This regularisation conflict is further compounded by self-information dilution under high-order propagation. For the Bernstein basis, $B_{k,K}(0) = 0$ for all $k \geq 1$, so higher-order terms assign zero self-weight at the zero-eigenvalue component, and for intermediate eigenvalues the self-contribution decreases monotonically with propagation order. In heterophilic graphs, where neighbours typically carry conflicting class signals, this attenuation of self-information compounds with the aggregation of high-frequency noise from cross-class neighbours, jointly degrading node-level discriminability. Empirical studies confirm that GNNs on strongly heterophilic graphs often perform no better than feature-only multi-layer perceptrons (MLPs) (Zhu et al., 2020; Lim et al., 2021; Li et al., 2018; Chen et al., 2020a), a failure that can be attributed to this combined effect. Section 3.1 addresses both aspects: the Morphological Component Analysis (MCA) decomposition resolves the regularisation conflict, and a learnable residual bypass counteracts self-information dilution.

2.5 Fixed Spectral Resolution

Resolving the regularisation conflict, however, does not address a second structural problem that is independent of the choice of regulariser. For the Bernstein basis, the k-th control point is anchored at $\lambda = 2k/K$, yielding $K+1$ uniformly spaced points over $[0, 2]$. When discriminative signals concentrate in a band of effective width $\delta = \lambda_{\text{hi}} - \lambda_{\text{lo}} < 2$, only $K_{\text{eff}} = \lfloor K\delta/2 \rfloor + 1$ control points cover the informative band; the remaining $K - K_{\text{eff}} \approx K(1 - \delta/2)$ fall in the uninformative complement. Scaling K by any constant factor increases both counts proportionally, leaving the wasted fraction $1 - \delta/2$ unchanged. This inefficiency is therefore structural: it persists regardless of the regularisation scheme and cannot be eliminated by increasing model capacity. Section 3.2 resolves this by introducing a learnable spectral warp that adaptively restricts how many of the $K+1$ coefficients have their characteristic response realised within the graph's effective frequency band.

3 BernNetII

Having identified these structural obstacles, we introduce **BernNetII**: Section 3.1 addresses the regularisation conflict and feature dilution via the MCA decomposition and residual bypass, and Section 3.2 addresses the fixed distribution of control points via the learnable spectral warp. Proof sketches are given inline; complete proofs are in the appendix.

3.1 Coefficient Decomposition with Curvature Alignment

As established in Section 2, applying a single regulariser to a unified coefficient vector simultaneously degrades both the smooth and step components of the target filter. We resolve this by explicitly separating the two components at the parameter level.

Definition 2 (MCA synthesis parameterisation)**.** *The filter coefficient vector is defined as the sum of two independent learnable variables,*

$$\boldsymbol{\theta} := \boldsymbol{\alpha} + \boldsymbol{\beta}, \qquad \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{K+1}, \tag{4}$$

where $\boldsymbol{\alpha}$ is the smooth component and $\boldsymbol{\beta}$ is the step component.

The forward pass depends on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ only through their sum, so the data-fitting cross-entropy loss $\mathcal{L}_{CE}$ is identical to that of a single-vector baseline with the same $\boldsymbol{\theta}$; the inference parameter count is unchanged, as $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are summed into $\boldsymbol{\theta}$ after training; training maintains two vectors of size $K+1$, adding $\mathcal{O}(K)$ memory overhead. The three regularisation terms are

$$R_\alpha(\boldsymbol{\alpha}) = \mu \|\Delta^2 \boldsymbol{\alpha}\|_2^2 = \mu \sum_{k=0}^{K-2} (\alpha_{k+2} - 2\alpha_{k+1} + \alpha_k)^2, \tag{5}$$

$$R_\beta(\boldsymbol{\beta}) = \gamma \|\boldsymbol{\beta}\|_{DTV} = \gamma \sum_{k=0}^{K-1} |\beta_{k+1} - \beta_k|, \tag{6}$$

$$R_{\mathrm{ca}}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \nu \sum_{k=0}^{K} \frac{\beta_k^2}{|(\Delta^2 \boldsymbol{\alpha})_k^\dagger| + \varepsilon}, \tag{7}$$

where $(\Delta^2 \boldsymbol{\alpha})_k^\dagger$ is the boundary-padded second difference, $\mu, \gamma, \nu > 0$ are hyperparameters, and $\varepsilon > 0$ ensures numerical stability. The denominator weights in R_{ca}, defined as $w_k = (|(\Delta^2 \boldsymbol{\alpha})_k^\dagger| + \varepsilon)^{-1}$, are computed with stop-gradient before each update. As a result, R_{ca} does not contribute to the gradient with respect to $\boldsymbol{\alpha}$ and reduces to a weighted ℓ_2 penalty on $\boldsymbol{\beta}$ with fixed coefficients. This choice avoids an undesirable interaction: without stop-gradient, the gradient would encourage $\boldsymbol{\alpha}$ to increase local curvature to decrease w_k and reduce the penalty on β_k, which conflicts with the objective of R_α.

The term R_α penalises curvature in $\boldsymbol{\alpha}$ and promotes slowly varying sequences. The term R_β is the Fused-LASSO penalty (Tibshirani et al., 2005; Tibshirani, 2011), which promotes piecewise-constant sequences with few transitions. The curvature alignment term R_{ca} suppresses β_k where $|(\Delta^2 \boldsymbol{\alpha})_k^\dagger| \approx 0$, since the denominator is small and the effective penalty weight is large, and releases β_k where $\boldsymbol{\alpha}$ is highly curved, since the weight is near zero. This position-adaptive mechanism guides the functional division explicitly, upgrading the implicit effect of R_α and R_β alone.

The complete pre-classification embedding incorporates an initial residual term to prevent feature dilution under high-order propagation:

$$Z = (1 - \tau) \sum_{k=0}^{K} (\alpha_k + \beta_k)\, H_{k,K}^{(s)} + \tau X, \tag{8}$$

with output $\hat{Y} = \mathrm{softmax}(ZW)$. The scalar $\tau = \sigma(r)$ is parameterised by a learnable logit r, initialised to a dataset-specific value r_0 (Appendix A, Tables 4-5); setting $\tau = 0$ recovers BernNet. Since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ enter (8) only through their sum, $\nabla_{\boldsymbol{\alpha}} \mathcal{L}_{CE} = \nabla_{\boldsymbol{\beta}} \mathcal{L}_{CE} =: \boldsymbol{g}$, so a single backward pass supplies gradients for both variables.

Proposition 3 (Implicit functional specialisation)**.** *Let the objective function be $F(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{L}_{\mathrm{CE}}(\boldsymbol{\alpha} + \boldsymbol{\beta}) + \mu \|\Delta^2 \boldsymbol{\alpha}\|_2^2 + \gamma \|\boldsymbol{\beta}\|_{DTV}$ with hyperparameters $\mu, \gamma > 0$. Suppose a parameter vector $\boldsymbol{\theta}$ can be additively decomposed into a smooth component and a step component, $\boldsymbol{\theta} = \boldsymbol{\theta}_s + \boldsymbol{\theta}_{\mathrm{step}}$. The trivial decomposition $(\boldsymbol{\theta}, \mathbf{0})$ is strictly suboptimal (i.e., $F(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{\mathrm{step}}) < F(\boldsymbol{\theta}, \mathbf{0})$) if the regularised jump amplitude $\|\Delta^2 \boldsymbol{\theta}_{\mathrm{step}}\|_2$ exceeds the following threshold:*

$$\|\Delta^2 \boldsymbol{\theta}_{\mathrm{step}}\|_2 > \frac{C + \sqrt{C^2 + 4\mu\gamma \|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}}}{2\mu}, \tag{9}$$

where $C := 2\mu \|\Delta^2 \boldsymbol{\theta}_s\|_2$.

Proof sketch. Since $\boldsymbol{\theta}$ is unchanged, $\mathcal{L}_{CE}$ contributes nothing to ΔF. Expanding $\|\Delta^2(\boldsymbol{\alpha}+\boldsymbol{\delta})\|_2^2$ yields a quadratic term in $\|\Delta^2\boldsymbol{\delta}\|_2^2$ that grows faster than the DTV change, which is bounded linearly by the triangle inequality. Setting $\boldsymbol{\delta} = -\boldsymbol{\theta}_{\text{step}}$ and bounding the resulting cross term via Cauchy–Schwarz shows that $\Delta F < 0$ whenever the jump amplitude of $\boldsymbol{\theta}_{\text{step}}$ is sufficiently large relative to γ/μ. This shows that, provided the jump amplitude is large enough relative to γ/μ, the trivial decomposition $(\boldsymbol{\theta}, \mathbf{0})$ is not optimal, and the regularisers implicitly drive the decomposition toward the true structural split. Full proof in Appendix B. □

3.2 Learnable Spectral Warp

The MCA decomposition resolves the regularisation conflict, but leaves a second structural problem untouched: the Bernstein control points remain uniformly distributed over $[0, 2]$, regardless of where the graph's discriminative signals actually concentrate. We now address this directly.

We introduce a scalar parameter $v \in \mathbb{R}$ and set $s = \sigma(v) \in (0, 1)$. The warped propagation operators $P_1^{(s)} := \frac{s}{2}\tilde{L}$ and $P_2^{(s)} := I - P_1^{(s)}$ are obtained from the standard operators by scalar multiplication, so no additional sparse matrix multiplications are required. Substituting $P_1^{(s)}, P_2^{(s)}$ into the Pascal-triangle recurrence of BernNet (He et al., 2021) yields warped state matrices $H_{k,n}^{(s)}$ and the warped filter

$$g_s(\lambda) = \sum_{k=0}^{K} \theta_k B_{k,K}\left(\frac{s\lambda}{2}\right), \qquad \lambda \in [0, 2]. \tag{10}$$

As λ ranges over $[0, 2]$, the warped argument $s\lambda/2$ sweeps only $[0, s] \subset [0, 1]$, so only coefficients with $k \leq sK$ have their peak realised inside $[0, 2]$, rather than every control point relocating to $\lambda = 2sk/K$. The learned scalar s thus governs how many of the $K+1$ coefficients are effectively activated. Gradient descent on $\mathcal{L}_{CE}$ adapts s to the graph's effective spectral band, removing the need for manual bandwidth selection.

Proposition 4 (Warped DTV bound)**.** *For any $s \in (0, 1)$, $\|g_s\|_{TV} \leq \|\boldsymbol{\theta}\|_{DTV}$. The bound is independent of s.*

Proof sketch. Differentiating (10) gives $g_s'(\lambda) = \frac{sK}{2}\sum_{k=0}^{K-1}(\theta_{k+1} - \theta_k)B_{k,K-1}(s\lambda/2)$. Integrating over $[0, 2]$ with the substitution $u = s\lambda/2$ and applying $\int_0^1 B_{k,K-1}(u)du = 1/K$, the factor of s from the chain rule cancels with the Jacobian, yielding $\|g_s\|_{TV} \leq \|\boldsymbol{\theta}\|_{DTV}$. The complete derivation is in Appendix C. □

Proposition 4 shows that the warp reduces approximation error by focusing polynomial degrees on the effective band, while leaving the estimation error controlled by $\|\boldsymbol{\theta}\|_{DTV}$ unchanged. This separation parallels the advantage ChebNetII gains through optimal interpolation nodes.

3.3 Learning Objective and Optimisation

The full learning objective is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},W,v,\tau} \underbrace{\mathcal{L}_{CE} + R_\alpha + R_{\text{ca}}}_{h\text{: smooth}} + \underbrace{\gamma\|\boldsymbol{\beta}\|_{DTV}}_{r\text{: non-smooth}}. \tag{11}$$

The curvature alignment term R_{ca} enters the objective with stop-gradient applied to the denominator weights $w_k = (|(\Delta^2\boldsymbol{\alpha})_k^\dagger| + \varepsilon)^{-1}$. Consequently, when optimising $\boldsymbol{\alpha}$, R_{ca} contributes only through its $\boldsymbol{\beta}$-dependent numerator, and the denominator is treated as a fixed scalar; h is therefore smooth with respect to $\boldsymbol{\alpha}$. When optimising $\boldsymbol{\beta}$, the weights w_k are fixed positive scalars, so R_{ca} is a weighted ℓ_2 term, which is also smooth. The smooth part h in (11) is therefore L_h-smooth with respect to all variables simultaneously, and Theorem 5 applies.

We minimise (11) by joint proximal gradient descent, updating all variables simultaneously from a single shared backward pass rather than sequentially as in block coordinate descent. The update rules

are

$$W^{(t+1)} = W^{(t)} - \eta_W \nabla_W h^{(t)}, \tag{12}$$

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \eta_\alpha \big(\boldsymbol{g}^{(t)} + 2\mu(\Delta^2)^\top \Delta^2 \boldsymbol{\alpha}^{(t)}\big), \tag{13}$$

$$\boldsymbol{\beta}^{(t+1)} = \mathrm{prox}_{\eta_\beta \gamma \|\cdot\|_{DTV}}\big(\boldsymbol{\beta}^{(t)} - \eta_\beta \boldsymbol{g}^{(t)}\big). \tag{14}$$

The proximal operator in (14) reduces to a one-dimensional Fused-LASSO problem, solvable in $\mathcal{O}(K)$ time by the algorithm of Johnson (2013). In practice W, $\boldsymbol{\alpha}$, v, and τ are updated with Adam (Kingma & Ba, 2014) to accelerate convergence; $\boldsymbol{\beta}$ uses a fixed step size $\eta_\beta \leq 1/L_\beta$ (where L_β is the Lipschitz constant of $\nabla_{\boldsymbol{\beta}} h$) to maintain the convergence guarantee below. The theoretical result applies to the fixed-step scheme; using Adam for the smooth variables is a standard engineering practice with no effect on the guarantee for $\boldsymbol{\beta}$.

Theorem 5 (Joint Proximal Gradient Descent (PGD) Convergence)**.** *Consider the idealised fixed-step scheme in which all variables $\mathbf{u} := (W, \boldsymbol{\alpha}, v, \tau, \boldsymbol{\beta})$ are updated by joint proximal gradient descent with a uniform step size $\eta \leq 1/L_h$, where L_h is the global Lipschitz constant of ∇h on the joint space. Then the sequence $\{F^{(t)}\}$ is monotonically non-increasing and converges, and every limit point satisfies the first-order optimality conditions for all variables.*

Proof sketch. For smooth variables the descent lemma (Beck & Teboulle, 2009) gives $h^{(t+1)} \leq h^{(t)} - \frac{1}{2\eta}\|\Delta x^{(t)}\|^2$. For $\boldsymbol{\beta}$ the proximal gradient descent lemma gives

$$h(\boldsymbol{\beta}^{(t+1)}) + r(\boldsymbol{\beta}^{(t+1)}) \leq h(\boldsymbol{\beta}^{(t)}) + r(\boldsymbol{\beta}^{(t)}) - \frac{1}{2\eta_\beta}\|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|^2. \tag{15}$$

Summing the updates gives $\mathcal{F}^{(t+1)} \leq \mathcal{F}^{(t)}$. Since $\mathcal{F}$ is non-negative, the sequence converges. The optimality conditions follow from the vanishing differences between successive iterates. A complete proof is provided in Appendix D. □

Theorem 5 establishes convergence for fixed step gradient descent applied to the smooth variables. In practice, W, $\boldsymbol{\alpha}$, v, and τ are updated using Adam (Kingma & Ba, 2014), which improves convergence speed on large graphs. Adam does not satisfy the descent conditions required by Theorem 5 (Reddi et al., 2018), so the result does not formally apply to these variables. The convergence guarantee therefore applies to the proximal update of $\boldsymbol{\beta}$, which uses a fixed step size $\eta_\beta \leq 1/L_\beta$ throughout training. For the remaining variables, convergence is evaluated empirically using the training loss curves in Appendix D. This separation between a theoretical guarantee for the non-smooth update and empirical validation for the remaining variables is consistent with prior work, including BernNet (He et al., 2021) and ChebNetII (He et al., 2022).

Proposition 6 (Generalisation bound)**.** *Assume that the node features are bounded by $\|x_i\|_2 \leq R$, and that the spectral norm of the warped basis matrices satisfies $\|B_{k,K}(P_1^{(s)})\|_2 \leq 1$ for all k. Let the output function be parameterized by $Z(v,:) = \sum_{k=0}^{K}(\alpha_k + \beta_k)H_{k,K}(v,:)$, where $\boldsymbol{\theta} = \boldsymbol{\alpha} + \boldsymbol{\beta}$. Define the hypothesis class:*

$$\mathcal{H}_{C,B,M} := \{v \mapsto Z(v,:) \mid \|\boldsymbol{\beta}\|_{DTV} \leq C,\ \|\boldsymbol{\alpha}\|_\infty \leq B,\ |\alpha_0 + \beta_0| \leq M\}. \tag{16}$$

Then, the empirical Rademacher complexity satisfies:

$$\hat{\mathfrak{R}}_n(\mathcal{H}_{C,B,M}) \leq \frac{R}{\sqrt{n}}\left(M + CK\sqrt{2\log(2K)} + 2BK^2\sqrt{2\log(2K)}\right). \tag{17}$$

Proof sketch. Abel summation separates the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ contributions; Massart's finite-class lemma (Massart, 2000) bounds each term independently. Full derivation in Appendix E. □

The bound shows that the MCA decomposition controls model complexity through C for the step component and B for the smooth component; tightening either constraint reduces the generalisation gap independently of the other. The complete algorithm is given in Appendix G.

Table 1: Dataset statistics.

	Chameleon	Squirrel	Actor	Texas	Cornell	Cora	Citeseer	Pubmed
Nodes	2,277	5,201	7,600	183	183	2,708	3,327	19,717
Edges	31,371	198,353	26,659	279	277	5,278	4,552	44,324
Features	2,325	2,089	932	1,703	1,703	1,433	3,703	500
Classes	5	5	5	5	5	7	6	3
$\mathcal{H}(\mathcal{G})$	0.23	0.22	0.22	0.11	0.30	0.81	0.74	0.80

Table 2: Mean classification accuracy of full-supervised node classification with random splits. The best result in each column is shown in bold, and the second best is underlined.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
MLP	46.59±1.84	31.01±1.18	86.81±2.24	84.15±3.05	40.18±0.55	76.89±0.97	76.52±0.89	86.14±0.25
GCN	60.81±2.95	45.87±0.88	76.97±3.97	65.78±4.16	33.26±1.15	87.18±1.12	79.85±0.78	86.79±0.31
ChebNet	59.51±1.25	40.81±0.42	86.28±2.62	83.91±2.17	37.42±0.58	87.32±0.92	79.33±0.57	87.82±0.24
ARMA	60.21±1.00	36.27±0.62	83.97±3.77	85.62±2.13	37.67±0.54	87.13±0.80	80.04±0.55	86.93±0.24
APPNP	52.15±1.79	35.71±0.78	90.64±1.70	91.52±1.81	39.76±0.49	88.16±0.74	80.47±0.73	88.13±0.33
GCNII	63.44±0.85	41.96±1.02	80.46±5.91	84.26±2.13	36.89±0.95	88.46±0.82	79.97±0.65	**89.94**±0.31
TWIRLS	50.21±2.97	39.63±1.02	91.31±3.36	89.83±2.29	38.13±0.81	88.57±0.91	80.07±0.94	88.87±0.43
EGNN	51.55±1.73	35.81±0.91	81.34±1.56	82.09±1.16	35.16±0.64	87.47±1.33	80.51±0.93	88.74±0.46
PDE-GCN	66.01±1.56	48.73±1.06	93.24±2.03	89.73±1.35	39.76±0.74	88.62±1.03	79.98±0.97	89.92±0.38
GPR-GNN	67.49±1.38	50.43±1.89	92.91±1.32	91.57±1.96	39.91±0.62	88.54±0.67	80.13±0.84	88.46±0.31
BernNet	68.53±1.68	51.39±0.92	92.62±1.37	92.13±1.64	41.71±1.12	88.51±0.92	80.08±0.75	88.51±0.39
ChebNetII	71.37±1.01	**57.72**±0.59	93.28±1.47	92.30±1.48	41.75±1.07	**88.71**±0.93	80.53±0.79	88.93±0.29
BernNetII	**73.13**±1.44	55.75±0.64	**93.61**±1.97	**92.62**±2.13	**41.98**±0.35	88.13±0.71	**80.60**±0.78	89.07±0.35

4 EXPERIMENTS

We evaluate BernNetII on the node classification benchmarks used by ChebNetII (He et al., 2022), covering both semi-supervised and full-supervised random-split settings. The datasets include three homophilic citation graphs, Cora, Citeseer, and Pubmed (Sen et al., 2008; Yang et al., 2016); two Wikipedia graphs, Chameleon and Squirrel (Rozemberczki et al., 2021); and three heterophilic graphs from WebKB/Geom-GCN, Actor, Texas, and Cornell (Pei et al., 2020). For each graph, we report the edge homophily ratio $\mathcal{H}_{\mathcal{G}} = |\{(u, v) \in \mathcal{E} : y_u = y_v\}|/|\mathcal{E}|$ following Zhu et al. (2020), where y_v denotes the label of node v. The dataset statistics are summarized in Table 1. All experiments are run on a workstation with two NVIDIA RTX 3090 GPUs, an Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz, and 128GB RAM. For fair comparison, the baseline results in Tables 2 and 3 are taken from ChebNetII (He et al., 2022), and BernNetII is evaluated under the same data splits and reporting protocol. We report mean accuracy and standard deviation over ten random splits.

4.1 FULL-SUPERVISED NODE CLASSIFICATION

For full-supervised node classification, nodes are randomly split into $60\%/20\%/20\%$ training, validation, and test sets, and all methods are evaluated on the same ten splits. This split convention follows the evaluation setting adopted by ChebNetII (He et al., 2022), which in turn follows prior full-supervised evaluations on these benchmarks (Pei et al., 2020; Chien et al., 2020; He et al., 2021). We compare against the polynomial-filter baselines used in the semi-supervised setting, including GCN (Kipf & Welling, 2016), ChebNet (Defferrard et al., 2016), ARMA (Bianchi et al., 2021), APPNP (Klicpera et al., 2019), GPR-GNN (Chien et al., 2020), and BernNet (He et al., 2021). We additionally include four competitive full-supervised methods: GCNII (Chen et al., 2020b), TWIRLS (Yang et al., 2021), EGNN (Zhou et al., 2021), and PDE-GCN (Eliasof et al., 2021). BernNetII uses a two-layer MLP with 64 hidden units and polynomial order $K = 10$ for all datasets. The smooth variables are trained with Adam, and the nonsmooth step component is updated by the fused-lasso proximal step described in Section 3.3; training uses early stopping with patience 200 and at most 1000 epochs. Detailed hyperparameters are reported in Appendix A. Table 2 reports the full-supervised results. Component ablation results are reported in Appendix F.

BernNetII achieves the best result on five out of eight datasets and the second-best result on Squirrel. The improvements are concentrated on heterophilic graphs, where the desired spectral response

Table 3: Mean classification accuracy of semi-supervised node classification with random splits. The best result in each column is shown in bold, and the second best is underlined.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
MLP	26.36±2.85	21.42±1.50	32.42±9.91	36.53±7.92	29.75±0.95	57.17±1.34	56.75±1.55	70.52±2.01
GCN	38.15±3.77	<u>31.18</u>±0.93	34.68±9.07	32.36±8.55	22.74±2.37	79.19±1.37	69.71±1.32	78.81±0.84
ChebNet	37.15±1.49	26.55±0.46	36.35±8.90	28.78±4.85	26.58±1.92	78.08±0.86	67.87±1.49	73.96±1.68
ARMA	37.42±1.72	24.15±0.93	39.65±8.09	28.90±10.07	27.02±2.31	79.14±1.07	69.35±1.44	78.31±1.33
APPNP	32.73±2.31	24.50±0.89	34.79±10.11	34.85±9.71	29.74±1.04	82.39±0.68	69.79±0.92	**79.97**±1.58
GPR-GNN	33.03±1.92	24.36±1.52	33.98±11.90	38.95±12.36	28.58±1.81	82.37±0.91	69.22±1.27	79.28±2.25
BernNet	27.32±4.04	22.37±0.98	43.01±7.45	39.42±9.59	<u>29.87</u>±0.78	82.17±0.86	69.44±0.97	79.48±1.47
ChebNetII	<u>43.42</u>±3.54	**33.96**±1.22	<u>46.58</u>±7.68	<u>42.19</u>±11.61	**30.18**±0.81	<u>82.42</u>±0.64	<u>69.89</u>±1.21	<u>79.51</u>±1.03
BernNetII	**43.68**±1.34	30.71±0.93	**54.62**±2.77	**49.31**±9.31	28.03±1.25	**82.96**±0.98	**69.98**±1.53	77.71±1.42

often contains both smooth structure and abrupt high-frequency changes. Compared with ChebNetII, BernNetII improves accuracy on Chameleon, Texas, Cornell, and Actor (Table 2). These gains are consistent with the proposed design: the spectral warp reallocates Bernstein resolution to the informative part of the spectrum, while the MCA decomposition allows smooth and step-like components to be regularised separately. On Squirrel and Cora, ChebNetII remains stronger, suggesting that Chebyshev interpolation can be advantageous when the target response is less localized or when global approximation accuracy dominates. On Pubmed, GCNII and PDE-GCN obtain the top two results, while BernNetII remains competitive with other spectral polynomial methods.

4.2 Semi-Supervised Node Classification

The semi-supervised setting uses the sparse-label protocol adopted by ChebNetII (He et al., 2022). For Cora, Citeseer, and Pubmed, we use the standard split (Yang et al., 2016): 20 labeled nodes per class for training, 500 nodes for validation, and 1000 nodes for testing. For Chameleon, Squirrel, Texas, Cornell, and Actor, we use sparse random splits with $2.5\%/2.5\%/95\%$ nodes for training, validation, and testing, respectively. We compare BernNetII with MLP, GCN (Kipf & Welling, 2016), ChebNet (Defferrard et al., 2016), ARMA (Bianchi et al., 2021), APPNP (Klicpera et al., 2019), GPR-GNN (Chien et al., 2020), BernNet (He et al., 2021), and ChebNetII (He et al., 2022). BernNetII uses the same training setup as above, with task-specific hyperparameters reported in Appendix A. Table 3 summarizes the semi-supervised results.

BernNetII obtains the best accuracy on five datasets in the sparse-label setting – Chameleon, Texas, Cornell, Cora, and Citeseer – with the most pronounced gains on Texas and Cornell, where it exceeds ChebNetII by 8.04 and 7.12 percentage points, respectively. These results suggest that separating the smooth and step components of the spectral filter is particularly useful when supervision is scarce: the smooth component stabilizes estimation, while the step component retains the ability to capture abrupt class-discriminative spectral transitions. BernNetII is weaker on Squirrel, Actor, and Pubmed, indicating that the current balance between curvature, fused-lasso, and residual terms may not be uniformly optimal across all graph regimes.

5 Conclusion

We introduced BernNetII, a Bernstein spectral GNN for heterophilic node classification that addresses two limitations of existing polynomial spectral models – fixed spectral resolution and homogeneous coefficient regularisation – through a learnable spectral warp and an MCA decomposition of the filter coefficients into smooth and step components, complemented by a learnable residual bypass that preserves node-level information under high-order propagation.

Empirically, BernNetII achieves strong performance across eight node classification benchmarks, with particularly clear gains on heterophilic datasets in both full-supervised and semi-supervised settings. A limitation is that the MCA decomposition introduces three hyperparameters μ, γ, and ν, whose optimal values may vary across graph regimes; the current approach relies on grid search and does not provide a principled selection rule.

AI USE STATEMENT

In this work, we used generative AI tools solely to polish the language of the manuscript, including improving grammar, wording, and clarity of sentences drafted by the authors. We did not use generative AI tools for research ideation, experimental design, the development of the proposed method, theoretical derivations or proofs, code implementation, data analysis, or the generation of any results, figures, or tables reported in this paper; all such tasks were carried out entirely by the authors. All AI-assisted language edits were reviewed by the authors to confirm that they preserved the intended scientific meaning and introduced no factual, technical, or numerical changes. We take responsibility for the final content of this work, including all text, claims, and results.

ETHICS STATEMENT

This work uses only standard, publicly available graph node-classification benchmarks (Cora, Citeseer, Pubmed, Chameleon, Squirrel, Actor, Texas, and Cornell) and does not involve human subjects, crowdsourcing, or the collection of new or sensitive data. All baseline methods and datasets are properly cited to their original sources. We do not foresee negative societal impacts beyond the general risks associated with methodological research on graph neural networks, and no additional safeguards beyond standard research practice are required for the released code.

REPRODUCIBILITY STATEMENT

All information needed to reproduce the main results is provided in the paper and its appendix. Sections 3 and 3.3 give the complete model definition, regularisation terms, and optimisation updates; Appendix A reports the full hyperparameter settings (regularisation weights, initial warp and residual values, and dataset splits) for both the full-supervised and semi-supervised settings; Appendices B, C, D, and E give the complete assumptions and proofs for all theoretical results (Theorem 1, Proposition 3, Proposition 4, Theorem 5, and Proposition 6); and Appendix G gives the full training procedure as pseudocode. Code implementing BernNetII, including scripts to reproduce the reported experiments, is publicly available at `https://github.com/AI-FZY/BernNetII`.

REFERENCES

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3496–3507, 2021.

Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3438–3445, 2020a.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735, 2020b.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Moshe Eliasof, Eldad Haber, and Eran Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Rida T Farouki. The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6):379–419, 2012.

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.

Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in neural information processing systems*, 34:14239–14251, 2021.

Mingguo He, Zhewei Wei, and Ji-Rong Wen. Convolutional neural networks on graphs with chebyshev approximation, revisited. *Advances in neural information processing systems*, 35: 7264–7276, 2022.

Nicholas A Johnson. A dynamic programming algorithm for the fused lasso and l 0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in neural information processing systems*, 34:20887–20902, 2021.

George G Lorentz. *Bernstein polynomials*. American Mathematical Soc., 2012.

John C Mason and David C Handscomb. *Chebyshev polynomials*. Chapman and Hall/CRC, 2002.

Pascal Massart. About the constants in talagrand's concentration inequalities for empirical processes. *The Annals of Probability*, 28(2):863–884, 2000.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2110–2119, 2018.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(1):91–108, 2005.

Ryan J Tibshirani. *The solution path of the generalized lasso*. Stanford University, 2011.

Table 4: Hyperparameters of BernNetII for full-supervised node classification.

Dataset	layer	hidden	K	lr_l	$L_{2,l}$	$\mathrm{dropout}_l$	lr_p	$L_{2,p}$	$\mathrm{dropout}_p$	μ	γ	ν	v_0	r_0
Cora	2	64	10	0.01	0.0005	0.5	0.02	0.0	0.2	0.05	0.01	0.01	0.5	-3
Citeseer	2	64	10	0.01	0.0015	0.5	0.01	0.0	0.3	0.01	0.01	0.01	0.5	0
Pubmed	2	64	10	0.01	0.0	0.5	0.04	0.0	0.0	0.01	0.01	0.01	0.5	0
Chameleon	2	64	10	0.05	0.0	0.5	0.015	0.0	0.65	0.0001	0.0001	0.01	1	1
Squirrel	2	64	10	0.05	0.0	0.5	0.02	0.0005	0.6	0.0001	0.0001	0.01	0.5	-6
Actor	2	64	10	0.05	0.0	0.5	0.01	0.0	0.9	0.0001	0.0001	0.01	1	1
Texas	2	64	10	0.05	0.0005	0.5	0.001	0.0	0.6	0.3	0.001	0.01	1	-6
Cornell	2	64	10	0.05	0.0005	0.5	0.001	0.0005	0.7	0.3	0.1	0.01	0.5	-6

Table 5: Hyperparameters of BernNetII for semi-supervised node classification with random splits.

Dataset	layer	hidden	K	lr_l	$L_{2,l}$	$\mathrm{dropout}_l$	lr_p	$L_{2,p}$	$\mathrm{dropout}_p$	μ	γ	ν	v_0	r_0
Cora	2	64	10	0.01	0.0005	0.5	0.01	0.0	0.8	0.05	0.01	0.01	0.5	-3
Citeseer	2	64	10	0.01	0.0015	0.5	0.005	0.0	0.8	0.05	0.01	0.01	0.5	-3
Pubmed	2	64	10	0.01	0.0	0.5	0.005	0.0	0.8	0.05	0.01	0.01	0.5	-3
Chameleon	2	64	10	0.01	0.0	0.5	0.0005	0.0	0.5	0.001	0.0001	0.0	1.0	-6
Squirrel	2	64	10	0.01	0.0005	0.5	0.0005	0.0	0.5	0.001	0.0001	0.0	1.0	-6
Actor	2	64	10	0.05	0.0	0.5	0.01	0.0	0.6	0.0001	0.0001	0.01	1	1
Texas	2	64	10	0.05	0.0005	0.8	0.001	0.0	0.7	0.3	0.001	0.01	1	-6
Cornell	2	64	10	0.05	0.0005	0.8	0.001	0.0005	0.7	0.3	0.1	0.01	0.5	-6

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 346–353, 2019.

Yongyi Yang, Tang Liu, Yangkun Wang, Jinjing Zhou, Quan Gan, Zhewei Wei, Zheng Zhang, Zengfeng Huang, and David Wipf. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning*, 2021.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pp. 40–48, 2016.

Kaixiong Zhou, Xiao Huang, Daochen Zha, Rui Chen, Li Li, Soo-Hyun Choi, and Xia Hu. Dirichlet energy constrained learning for deep graph neural networks. *Advances in Neural Information Processing Systems*, 34:21834–21846, 2021.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

A Hyperparameters

BernNetII uses the same two-layer MLP backbone and polynomial order for all datasets, but introduces additional model-specific quantities through the MCA decomposition, spectral warp, and residual path. We therefore report the regularisation weights for the smooth coefficient component $\boldsymbol{\alpha}$ and the step component $\boldsymbol{\beta}$, together with the initial values of the learnable warp and residual parameters. Here μ is the ℓ_2 second-difference regularisation weight for $\boldsymbol{\alpha}$, γ is the TV regularisation strength on $\boldsymbol{\beta}$, and ν is the curvature-alignment regularisation weight. The column v_0 denotes the initial value of the learnable spectral-warp parameter, and r_0 denotes the initial residual-path logit, with the residual weight given by $\tau = \sigma(r)$. These initial values provide reproducible starting points for graph-dependent adaptation; both parameters are learned during training and do not fix the final spectral bandwidth or residual strength. The coefficient-energy weight is fixed to $\eta = 10^{-5}$ in all experiments and is therefore omitted from the tables.

B Proof of Proposition 3 (Implicit Functional Specialisation)

Proof. Consider the difference in the total objective cost between the trivial decomposition $(\boldsymbol{\theta}, \mathbf{0})$ and the separated decomposition $(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{\mathrm{step}})$. Since $\boldsymbol{\theta} = \boldsymbol{\theta}_s + \boldsymbol{\theta}_{\mathrm{step}}$ by definition, the cross-entropy loss

$\mathcal{L}_{\mathrm{CE}}$ remains identical. The objective reduction, denoted as $\Delta F_{\mathrm{gain}} := F(\boldsymbol{\theta}, \mathbf{0}) - F(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{\mathrm{step}})$, is determined entirely by the change in regularisation penalties. Specifically, the DTV penalty changes by $\gamma(\|\mathbf{0}\|_{DTV} - \|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}) = -\gamma\|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}$, yielding:

$$\Delta F_{\mathrm{gain}} = \mu\big(\|\Delta^2(\boldsymbol{\theta}_s + \boldsymbol{\theta}_{\mathrm{step}})\|_2^2 - \|\Delta^2\boldsymbol{\theta}_s\|_2^2\big) - \gamma\|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}. \tag{18}$$

Expanding the squared L_2 norm for the smooth penalty gives:

$$\|\Delta^2(\boldsymbol{\theta}_s + \boldsymbol{\theta}_{\mathrm{step}})\|_2^2 - \|\Delta^2\boldsymbol{\theta}_s\|_2^2 = \|\Delta^2\boldsymbol{\theta}_{\mathrm{step}}\|_2^2 + 2\langle\Delta^2\boldsymbol{\theta}_s, \Delta^2\boldsymbol{\theta}_{\mathrm{step}}\rangle. \tag{19}$$

Applying the Cauchy–Schwarz inequality to the inner product in (19), we establish the lower bound $2\langle\Delta^2\boldsymbol{\theta}_s, \Delta^2\boldsymbol{\theta}_{\mathrm{step}}\rangle \geq -2\|\Delta^2\boldsymbol{\theta}_s\|_2\|\Delta^2\boldsymbol{\theta}_{\mathrm{step}}\|_2$. Substituting this bound back into (18) gives a strict lower bound on the objective reduction:

$$\Delta F_{\mathrm{gain}} \geq \mu\|\Delta^2\boldsymbol{\theta}_{\mathrm{step}}\|_2^2 - 2\mu\|\Delta^2\boldsymbol{\theta}_s\|_2\|\Delta^2\boldsymbol{\theta}_{\mathrm{step}}\|_2 - \gamma\|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}. \tag{20}$$

Let $A := \|\Delta^2\boldsymbol{\theta}_{\mathrm{step}}\|_2$ and $C := 2\mu\|\Delta^2\boldsymbol{\theta}_s\|_2$. The right-hand side of (20) defines a convex quadratic function in A, given by $g(A) = \mu A^2 - CA - \gamma\|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}$.

Since the leading coefficient $\mu > 0$, the quadratic function $g(A)$ is strictly positive for any value of A that exceeds its largest real root. Solving $g(A) = 0$ via the quadratic formula yields the critical requirement:

$$A > \frac{C + \sqrt{C^2 + 4\mu\gamma\|\boldsymbol{\theta}_{\mathrm{step}}\|_{DTV}}}{2\mu}.$$

Under condition (9), we have $g(A) > 0$. By the transitivity of the established lower bound in (20), this guarantees $\Delta F_{\mathrm{gain}} \geq g(A) > 0$. Consequently, $F(\boldsymbol{\theta}_s, \boldsymbol{\theta}_{\mathrm{step}}) < F(\boldsymbol{\theta}, \mathbf{0})$, confirming that the trivial decomposition $(\boldsymbol{\theta}, \mathbf{0})$ is strictly suboptimal. □

C Proof of Proposition 4 (Warped TV Bound)

Proof. Differentiating (10) with respect to λ:

$$g_s'(\lambda) = \sum_{k=0}^{K} \theta_k \cdot \frac{d}{d\lambda} B_{k,K}\Big(\frac{s\lambda}{2}\Big) = \sum_{k=0}^{K} \theta_k \cdot \frac{s}{2} \cdot B_{k,K}'\Big(\frac{s\lambda}{2}\Big). \tag{21}$$

Applying the standard Bernstein derivative formula $B_{k,K}'(x) = K(B_{k-1,K-1}(x) - B_{k,K-1}(x))$ and re-indexing the resulting sum (setting $j = k - 1$ in the first term), we obtain

$$g_s'(\lambda) = \frac{sK}{2} \sum_{k=0}^{K-1} (\theta_{k+1} - \theta_k) B_{k,K-1}\Big(\frac{s\lambda}{2}\Big). \tag{22}$$

Taking absolute values and applying the triangle inequality:

$$|g_s'(\lambda)| \leq \frac{sK}{2} \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k|\, B_{k,K-1}\Big(\frac{s\lambda}{2}\Big). \tag{23}$$

(Non-negativity of $B_{k,K-1}$ permits removal of the absolute value from the basis function.) Integrating over $[0, 2]$ and let $u = s\lambda/2$, so $d\lambda = 2\,du/s$, limits $u \in [0, s] \subset [0, 1]$:

$$\begin{aligned}
\|g_s\|_{TV} &\leq \frac{sK}{2} \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k| \int_0^2 B_{k,K-1}\Big(\frac{s\lambda}{2}\Big) d\lambda \\
&= \frac{sK}{2} \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k| \cdot \frac{2}{s} \int_0^s B_{k,K-1}(u)\, du \\
&\leq \frac{sK}{2} \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k| \cdot \frac{2}{s} \int_0^1 B_{k,K-1}(u)\, du.
\end{aligned} \tag{24}$$

Using the Bernstein integral identity $\int_0^1 B_{k,n}(u)\,du = 1/(n+1)$ with $n = K-1$:

$$\|g_s\|_{TV} \le \frac{sK}{2} \cdot \frac{2}{s} \cdot \frac{1}{K} \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k| = \sum_{k=0}^{K-1} |\theta_{k+1} - \theta_k| = \|\boldsymbol{\theta}\|_{DTV}. \tag{25}$$

The factors s and $1/s$ cancel, making the bound independent of s, which completes the proof. □

D Proof of Theorem 5 (Joint PGD Convergence)

Proof. Let the objective be written as $F(\mathbf{u}) = h(\mathbf{u}) + r(\boldsymbol{\beta})$, where h is the smooth part with an L_h-Lipschitz continuous gradient with respect to the joint vector $\mathbf{u}$, and $r(\boldsymbol{\beta}) = \gamma\|\boldsymbol{\beta}\|_{DTV}$ is the non-smooth convex penalty.

Under the idealised scheme, all variables are updated simultaneously via a standard proximal gradient step in the joint vector space. Applying the proximal gradient descent lemma (Beck & Teboulle, 2009, Theorem 10.15) directly to $\mathbf{u}$ with step size $\eta \le 1/L_h$ yields the global sufficient decrease condition:

$$F(\mathbf{u}^{(t+1)}) \;\le\; F(\mathbf{u}^{(t)}) - \frac{1}{2\eta}\|\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}\|^2. \tag{26}$$

Expanding the squared L_2 norm into its component variables, we obtain:

$$\begin{aligned} F^{(t+1)} \;\le F^{(t)} - \frac{1}{2\eta}\Big(&\|W^{(t+1)} - W^{(t)}\|^2 + \|\boldsymbol{\alpha}^{(t+1)} - \boldsymbol{\alpha}^{(t)}\|^2 \\ &+ \|v^{(t+1)} - v^{(t)}\|^2 + \|\tau^{(t+1)} - \tau^{(t)}\|^2 + \|\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}\|^2\Big). \end{aligned} \tag{27}$$

This inequality establishes that $F^{(t+1)} \le F^{(t)}$. Since the objective function is bounded below ($F \ge 0$), the monotonically non-increasing sequence $\{F^{(t)}\}$ converges to a finite limit.

To establish first-order optimality, telescoping (27) from $t = 0$ to N and taking the limit as $N \to \infty$ gives:

$$\sum_{t=0}^{\infty} \|\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}\|^2 \;\le\; 2\eta\big(F^{(0)} - \lim_{t\to\infty} F^{(t)}\big) \;<\; \infty. \tag{28}$$

The convergence of the series implies that the successive differences for every individual variable vanish: $\|x^{(t+1)} - x^{(t)}\| \to 0$ for all $x \in \{W, \boldsymbol{\alpha}, v, \tau, \boldsymbol{\beta}\}$.

For the smooth variables $x \in \{W, \boldsymbol{\alpha}, v, \tau\}$, the exact update rule $x^{(t+1)} = x^{(t)} - \eta\nabla_x h^{(t)}$ dictates that $\nabla_x h^{(t)} = \frac{1}{\eta}(x^{(t)} - x^{(t+1)})$. Taking the limit as $t \to \infty$ yields $\nabla_x h^{(t)} \to \mathbf{0}$, satisfying the first-order optimality condition at every limit point.

For the proximal variable $\boldsymbol{\beta}$, the update rule inherently satisfies the subdifferential inclusion at iteration t:

$$\mathbf{0} \;\in\; \frac{1}{\eta}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^{(t+1)}) + \nabla_{\boldsymbol{\beta}} h^{(t)} + \gamma\,\partial\|\boldsymbol{\beta}^{(t+1)}\|_{DTV}. \tag{29}$$

As $t \to \infty$, we have $\frac{1}{\eta}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\beta}^{(t+1)}) \to \mathbf{0}$. By the continuous differentiability of h, $\nabla_{\boldsymbol{\beta}} h^{(t)} \to \nabla_{\boldsymbol{\beta}} h$. Because the subdifferential of a proper convex function has a closed graph, passing to the limit in (29) preserves the inclusion at the limit point:

$$\mathbf{0} \;\in\; \nabla_{\boldsymbol{\beta}} h + \gamma\,\partial\|\boldsymbol{\beta}\|_{DTV}, \tag{30}$$

which confirms the first-order optimality condition for $\boldsymbol{\beta}$. □

In practice, W, $\boldsymbol{\alpha}$, v, and τ are updated with Adam to accelerate convergence. Adam does not satisfy the fixed-step condition required by Theorem 5 (Reddi et al., 2018), so the theoretical convergence guarantee applies formally only to the proximal update of $\boldsymbol{\beta}$. Convergence of the remaining smooth variables under the empirical Adam scheme is validated empirically in Section 4.

E Proof of Proposition 6 (Generalisation Bound)

Proof. Without loss of generality, we bound the Rademacher complexity for a single output dimension d. The bound for the full output follows by the same argument applied coordinate-wise. Let $\{v_i\}_{i=1}^n$ be the labelled nodes and $\{\sigma_i\}_{i=1}^n$ be independent Rademacher variables. By definition of the empirical Rademacher complexity,

$$\hat{\mathfrak{R}}_n(\mathcal{H}) = \mathbb{E}_{\boldsymbol{\sigma}} \sup_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(v_i). \tag{31}$$

Applying Abel summation (summation by parts) to the projected output function $f(v_i) = \sum_{k=0}^{K} \theta_k [H_{k,K}(v_i, :)]_d$ gives:

$$\sum_{k=0}^{K} \theta_k [H_{k,K}(v_i, :)]_d = \theta_0 \Phi_0(v_i) + \sum_{j=0}^{K-1} (\theta_{j+1} - \theta_j) \Phi_{j+1}(v_i), \tag{32}$$

where $\Phi_0(v) := \sum_{k=0}^{K} [H_{k,K}(v, :)]_d$ and $\Phi_{j+1}(v) := \sum_{k=j+1}^{K} [H_{k,K}(v, :)]_d$ are the projected partial sums of the warped state matrices. Writing $\theta_{j+1} - \theta_j = d_j^\alpha + d_j^\beta$ with $d_j^\alpha := \alpha_{j+1} - \alpha_j$ and $d_j^\beta := \beta_{j+1} - \beta_j$, the right-hand side of (32) separates into three independent additive components. By the triangle inequality, the supremum in (31) is bounded by $T_0 + T_\beta + T_\alpha$, where each T collects the respective contributions of θ_0, $\{d_j^\beta\}$, and $\{d_j^\alpha\}$.

For T_0, the hypothesis class constraint guarantees $|\theta_0| = |\alpha_0 + \beta_0| \leq M$. Since $\{\sigma_i\}$ are independent with $\mathbb{E}[\sigma_i] = 0$ and $|\sigma_i| = 1$, by the Khintchine inequality and the norm bound $\|\Phi_0(v_i)\|_2 \leq R$, which follows from the node feature constraint and the partition of unity property of the Bernstein basis, yields:

$$T_0 \leq \frac{M}{n} \sqrt{\sum_{i=1}^{n} \|\Phi_0(v_i)\|_2^2} \leq \frac{M}{n} \sqrt{nR^2} = \frac{MR}{\sqrt{n}}. \tag{33}$$

For T_β, the DTV constraint yields $\sum_j |d_j^\beta| \leq \|\boldsymbol{\beta}\|_{DTV} \leq C$. Each $\Phi_{j+1}(v_i)$ is a partial sum of at most K projected warped state matrices, thus bounded by KR. By linearity, the supremum over the ℓ^1 ball $\left\{(d_j^\beta)_{j=0}^{K-1} : \sum_j |d_j^\beta| \leq C\right\}$ is achieved at its $2K$ extreme points, namely $\pm C\mathbf{e}_j$, where $\mathbf{e}_j$ is the standard basis vector. Applying Massart's finite-class lemma (Massart, 2000) to this finite set of $2K$ extreme points:

$$T_\beta \;\leq\; \frac{CKR\sqrt{2\log(2K)}}{\sqrt{n}}. \tag{34}$$

For T_α, the uniform bound $\|\boldsymbol{\alpha}\|_\infty \leq B$ implies $|d_j^\alpha| \leq 2B$ for all j, which entails the ℓ^1 relaxation $\sum_j |d_j^\alpha| \leq 2BK$. Evaluating the supremum over the relaxed ℓ^1 ball $\sum_j |d_j^\alpha| \leq 2BK$, the supremum is again achieved at its $2K$ extreme points $\pm 2BK\mathbf{e}_j$. Applying the same Massart argument yields:

$$T_\alpha \;\leq\; \frac{2BK \cdot KR\sqrt{2\log(2K)}}{\sqrt{n}} = \frac{2BK^2R\sqrt{2\log(2K)}}{\sqrt{n}}. \tag{35}$$

Summing the upper bounds (33), (34), and (35) directly yields (17), completing the proof. □

F Component Ablation Study

Table 6 reports one-component-removed ablations on all eight datasets under the full-supervised setting of Section 4.1, using the same splits and random seeds for every variant. The variants remove the spectral warp, coefficient decomposition, residual bypass, curvature-alignment penalty R_{ca}, or fused-lasso penalty R_β, respectively.

Table 6: Component ablation accuracies (%). Each column removes the indicated component from BernNetII. The best result in each row is bold.

Dataset	BernNetII	$-$warp	$-$decomp.	$-$residual	$-R_{\mathrm{ca}}$	$-R_\beta$
Chameleon	**73.13**	71.86	71.47	72.34	71.38	72.30
Squirrel	**55.75**	52.74	52.66	51.78	52.05	51.62
Texas	**93.61**	91.80	91.80	92.79	92.46	91.80
Cornell	**92.62**	91.48	92.46	90.49	91.48	90.49
Actor	41.98	41.56	41.82	42.62	**42.66**	41.77
Cora	**88.13**	87.32	87.72	87.06	87.59	87.52
Citeseer	80.60	80.85	80.60	**80.87**	80.63	80.74
Pubmed	**89.07**	88.88	87.98	87.87	88.98	88.27

The full model outperforms every ablated variant on six of the eight datasets. The largest and most consistent reductions occur on Squirrel: removing individual components lowers accuracy by 3.01–4.13 percentage points.

G Full Algorithm

Algorithm 1 BernNetII

Require: Graph $\mathcal{G}$, features X, labels Y, order K, hyperparameters μ, γ, ν, initial values v_0, r_0, step sizes $\eta_W, \eta_\alpha, \eta_\beta, \eta_v$

Ensure: $\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, W^*, v^*, \tau^*$

Precompute $(\Delta^2)^\top \Delta^2$ ▷ $\mathcal{O}(K)$, five-diagonal

Initialise $\boldsymbol{\alpha} \leftarrow \mathbf{1}/(K{+}1)$, $\boldsymbol{\beta} \leftarrow \mathbf{0}$, $v \leftarrow v_0$, $\tau \leftarrow \sigma(r_0)$

repeat

  $s \leftarrow \sigma(v)$; compute $\{H^{(s)}_{k,K}\}_{k=0}^{K}$ via Pascal recurrence ▷ $\mathcal{O}(K^2|\mathcal{E}|d)$

  $Z \leftarrow (1{-}\tau)\sum_k(\alpha_k{+}\beta_k)H^{(s)}_{k,K} + \tau X$

  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{CE}(ZW, Y)$ ▷ single backward pass; shared for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

  Update $W, \boldsymbol{\alpha}, v, \tau$ via Adam

  $\boldsymbol{\beta} \leftarrow \mathrm{prox}_{\eta_\beta \gamma \|\cdot\|_{DTV}}(\boldsymbol{\beta} - \eta_\beta \boldsymbol{g})$ ▷ Fused-LASSO, $\mathcal{O}(K)$

until convergence