GRAPH CONVOLUTION WITH DEGREE-POTENTIAL AND SPECTRAL CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) are widely used for node classification, aggregating neighborhood information through rules in which degree influences message weighting. Degree-aware methods have extended these rules through adaptive aggregation and structural augmentation. We investigate how relative source–receiver degrees can guide shared and local message calibration on the existing sparse graph. On PubMed, matching receiver-degree distributions reveals larger GCN–mean message changes in neighborhoods with greater degree contrast. This motivates *degree-potential calibrated graph convolution* (DPC-GCN), which combines shared and centered receiver-dependent corrections. To examine additional spectral flexibility after degree calibration, we introduce signed spectral calibration (SSC), yielding DPC-GCN-S. One shared scalar per layer learns the direction and strength of state adjustment, allowing preservation, smoothing, or sharpening. Our analysis explains the separation of shared and local corrections and shows that identical label spectral energies can admit opposite beneficial local calibration directions. Across 16 benchmarks, including three large-scale graphs, DPC-GCN matches or exceeds the best baseline mean on all eight homophilous datasets; DPC-GCN-S ranks first on four of five heterophilous datasets among the evaluated methods. An anonymized implementation is available at `https://anonymous.4open.science/r/DPCGCN-4ED8`.

1 INTRODUCTION

Graph neural networks (GNNs) are widely used for node classification, learning representations by aggregating neighborhood information (Gilmer et al., 2017). Representative models include GCN (Kipf & Welling, 2017) and GraphSAGE (Hamilton et al., 2017): GCN normalizes messages using both endpoint degrees, whereas the GraphSAGE mean aggregator averages over each neighborhood. Degree thus describes neighborhood size and participates in determining how neighboring information contributes to node representations. This role of degree has motivated aggregation designs that adapt to structural variation. PNA (Corso et al., 2020) combines degree-dependent scalers with multiple neighborhood statistics, while DegFairGNN (Liu et al., 2023) modulates aggregation with learnable contexts to address degree-related disparities. Subsequent analysis links degree bias to neighborhood homophily and diversity (Subramonian et al., 2024). More recently, DegFairGT (Hoang et al., 2025) combines structural augmentation and structural attention to mitigate such bias. These studies extend degree-aware modeling from aggregation rules to neighborhood structure.

These developments highlight degree's role in adaptive aggregation and structural modeling. We focus on its effect on individual normalized message weights: receivers with similar degrees can have different neighborhood degree profiles. We ask how relative source–receiver degrees should guide local calibration and how to separate it from a shared correction. To examine this variation, we compare symmetric GCN and uniform mean propagation on normalized PubMed features. After matching receiver-degree distributions, the weighted median relative message change increases from 0.098 to 0.334 across bins of increasing neighborhood degree contrast, measured by the mean absolute log-degree difference between receivers and their neighbors (Figure 1(a)). This observation motivates calibration that adapts to local degree relationships.

We therefore propose degree-potential calibration (DPC), parameterizing message corrections through the log-degree relation between these reference rules. The logarithm turns multiplicative degree ratios

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.

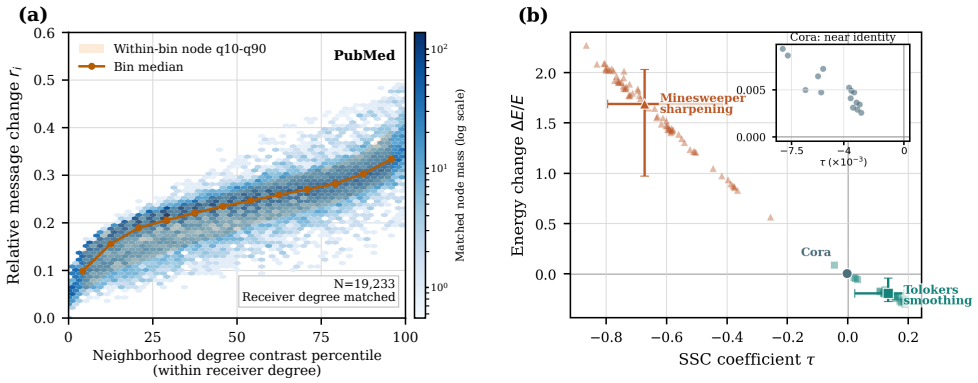

Figure 1: Degree-dependent message changes and learned state calibration. (a) GCN–mean message changes on PubMed increase with the mean absolute log-degree difference between receivers and their neighbors after matching receiver-degree distributions. Density, medians, and 10th–90th percentiles are weighted. (b) Learned SSC coefficients and energy responses in three diagnostic configurations. Points represent run–layer observations; bars show 10th–90th percentiles. Protocols are given in Appendix B.1.

into additive differences between endpoint potentials. A shared coefficient adjusts degree weighting across the graph, while a centered topology coordinate captures receiver-dependent deviations. Centering separates local corrections from shared mixing, so a nonzero local correction cannot be replaced by changing the shared coefficient alone (Proposition 1). The resulting DPC-GCN retains sparse propagation and recovers GCN and mean propagation as special cases.

These degree-based corrections also reshape the aggregated states, raising a further question: can directly adjusting their spectral response improve classification? Existing spectral methods provide flexible responses: AdaGNN (Dong et al., 2021) learns channel-specific filters, while CPF (Li et al., 2025) adapts filtering across graph partitions. After degree calibration, we investigate how much additional spectral flexibility is needed. Our analysis shows that identical label spectral energies can admit opposite beneficial local calibration directions (Proposition 3), motivating learning the calibration sign from classification supervision. We use signed spectral calibration (SSC), a first-order Laplacian update that learns direction and strength with one scalar per layer, yielding DPC-GCN-S. Matched controls on five heterophilous datasets support this compact choice, with no consistent gains from finer-grained coefficients (Appendix B.3). Figure 1(b) illustrates the learned behavior: Cora remains near identity, Minesweeper sharpens, and Tolokers smooths. Together, these calibrations adapt message weights and aggregated states within a sparse local architecture. Our contributions are:

- We propose DPC, which translates relative source–receiver degrees into explicit message-weight adjustments. Shared and centered receiver-dependent corrections adapt calibration to local degree contrasts while retaining sparse propagation.
- We study signed spectral calibration (SSC) of degree-calibrated states using one shared scalar per layer. Our analysis shows that identical label spectral energies and initial standardized-margin distributions can admit opposite beneficial local calibration directions, motivating learning the calibration sign from classification supervision.
- Across 16 benchmarks, DPC-GCN matches or exceeds the best baseline mean on all eight homophilous datasets, and DPC-GCN-S leads on four of five heterophilous datasets. Matched controls support the calibration designs and show no consistent advantage from finer-grained SSC parameterizations.

2 Background

Degree-aware propagation. Let $G = (V, E)$ be an undirected graph with $N = |V|$ nodes, adjacency A, and node features $X \in \mathbb{R}^{N \times d_0}$. We add one self-loop per node and define $\widetilde{A} = A + I$, $\widetilde{d}_i = \sum_j \widetilde{A}_{ij}$, and $\widetilde{D} = \mathrm{Diag}(\widetilde{d}_1, \ldots, \widetilde{d}_N)$. For transformed node states Z, weighted local

aggregation takes the form $B = PZ$, where P_{ij} determines the contribution of source node j to receiver i. Two standard choices are symmetric normalization in GCN (Kipf & Welling, 2017) and neighborhood averaging as used in the mean aggregator of GraphSAGE (Hamilton et al., 2017), giving

$$P_{\mathrm{g}} = \widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}, \qquad P_{\mathrm{m}} = \widetilde{D}^{-1}\widetilde{A}. \tag{1}$$

The symmetric operator P_{g} accounts for both endpoint degrees, whereas P_{m} assigns uniform weights within each receiver's neighborhood including the self-loop, with $P_{\mathrm{m}}\mathbf{1} = \mathbf{1}$. Changing normalization can therefore alter the relative contributions of neighbors with different degrees. The two operators satisfy the standard diagonal similarity relation $P_{\mathrm{m}} = \widetilde{D}^{-1/2}P_{\mathrm{g}}\widetilde{D}^{1/2}$ from normalized random-walk spectral theory (Chung, 1997); they share a spectrum but generally produce different messages from the same node state.

Degree-aware and adaptive aggregation extend how neighborhood structure enters message passing. PNA (Corso et al., 2020) combines multiple neighborhood statistics with degree-dependent scalers. Generalized aggregators (Li et al., 2020) learn flexible neighborhood summaries, while GAT (Veličković et al., 2018) and GATv2 (Brody et al., 2022) learn feature-dependent neighbor weights. Co-GNN (Finkelshtein et al., 2024) instead learns node-wise communication policies. These designs adapt neighborhood summaries or interactions at different levels. Our focus is the degree dependence of normalized edge weights, parameterized through source–receiver degree-potential contrasts.

Graph filtering. Following spectral graph theory and graph signal processing (Chung, 1997; Shuman et al., 2013), we define graph frequency and energy using the symmetric GCN reference operator:

$$\begin{aligned} L_{\mathrm{g}} &= I - P_{\mathrm{g}}, \\ \mathcal{E}(B) &= \frac{1}{2}\operatorname{Tr}(B^{\top}L_{\mathrm{g}}B) = \frac{1}{4}\sum_{i,j}\widetilde{A}_{ij}\left\|\frac{B_{i:}}{\sqrt{\widetilde{d}_i}} - \frac{B_{j:}}{\sqrt{\widetilde{d}_j}}\right\|_2^2. \end{aligned} \tag{2}$$

The graph-defined L_{g} remains fixed when the propagation operator is calibrated, providing a common reference for comparing node states. The pairwise energy measures variation between degree-normalized states across edges. The eigenvectors of L_{g} define graph-frequency modes: small eigenvalues correspond to low-frequency modes with little degree-normalized variation, and large eigenvalues to high-frequency modes with greater variation. In the spectral domain, this energy weights each mode's squared amplitude by its Laplacian eigenvalue.

Graph frequency also provides a framework for understanding how filtering and diffusion reshape node representations. Spectral GNNs parameterize responses in a graph Fourier basis (Bruna et al., 2014) or through polynomial (Defferrard et al., 2016) and rational (Levie et al., 2019; Bianchi et al., 2022) filters. Adaptive filters introduce flexibility through learned polynomial coefficients (Chien et al., 2021; He et al., 2021; Wang & Zhang, 2022), channel-specific responses (Dong et al., 2021), mixtures of low- and high-frequency channels (Bo et al., 2021; Luan et al., 2022), or partition-specific responses (Li et al., 2025). Neural diffusion relates representation updates to learnable graph dynamics (Chamberlain et al., 2021a;b; Eliasof et al., 2021), while work on multi-order smoothness studies its relation to generalization (Gu et al., 2025). We study a restricted first-order correction applied after degree-calibrated aggregation, with one coefficient per layer shared across nodes and features. Matched controls assess its incremental contribution and parameter granularity.

Graph Transformers. Graph Transformers extend global self-attention (Vaswani et al., 2017) to graph-structured inputs. One line injects topology through structural, positional, or spectral encodings (Ying et al., 2021; Kreuzer et al., 2021; Rampášek et al., 2022; Bo et al., 2023); another sparsifies or linearizes global interaction and combines it with local propagation (Wu et al., 2022; 2023a; Kong et al., 2023; Wu et al., 2023b; Deng et al., 2024; Zhuo et al., 2025).

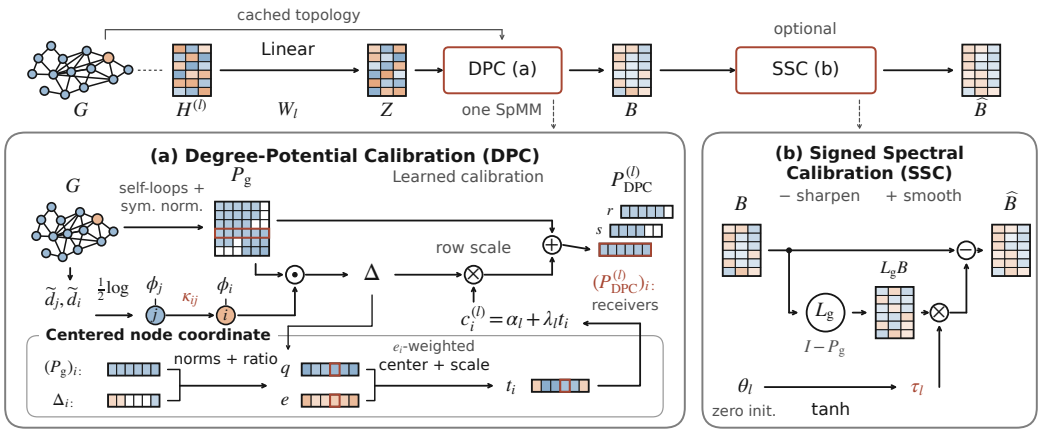

Figure 2: Overview of DPC-GCN and DPC-GCN-S. DPC combines a shared correction with a centered node-dependent correction to calibrate message weights; optional SSC applies a signed spectral correction to the aggregated state.

3 Degree-Potential Calibrated Graph Convolution

Degree-potential calibration (DPC) adapts message construction to local degree-potential contrasts through shared and centered node-wise corrections. Signed spectral calibration (SSC) then adjusts the aggregated state's graph-frequency response with one learnable signed scalar per layer.

Edge-wise degree-potential form. We parameterize message corrections through the source–receiver degree relation between symmetric GCN and mean propagation. On each effective edge $j \to i$,

$$\frac{(P_{\mathrm{m}})_{ij}}{(P_{\mathrm{g}})_{ij}} = \sqrt{\frac{\widetilde{d}_j}{\widetilde{d}_i}}. \tag{3}$$

For each node i, we define the *discrete log-degree potential* $\phi_i = \frac{1}{2}\log \widetilde{d}_i$. The logarithm converts the multiplicative degree ratio in Equation (3) into the additive directed contrast $\phi_j - \phi_i$ on edge $j \to i$. The resulting relative edge-weight change is

$$\kappa_{ij} := \exp(\phi_j - \phi_i) - 1 = \sqrt{\frac{\widetilde{d}_j}{\widetilde{d}_i}} - 1, \qquad (P_{\mathrm{m}})_{ij} = (P_{\mathrm{g}})_{ij}(1 + \kappa_{ij}). \tag{4}$$

Writing $\Delta = P_{\mathrm{m}} - P_{\mathrm{g}}$ gives $\Delta_{ij} = (P_{\mathrm{g}})_{ij}\kappa_{ij}$, the propagation correction calibrated by DPC. Appendix A.1 relates its magnitude to local degree-potential variation.

Layer overview. For layer l, transform input $H^{(l)}$ with trainable map W_l as $Z^{(l)} = H^{(l)}W_l$. Set $c_i^{(l)} = \alpha_l + \lambda_l t_i$, combining shared and node-wise coefficients through the topology coordinate t_i defined below. The propagation core (Figure 2) is

$$\begin{aligned} B_i^{(l)} &= \sum_j (P_{\mathrm{g}})_{ij}\left[1 + c_i^{(l)}\kappa_{ij}\right] Z_j^{(l)}, \\ \widehat{B}^{(l)} &= (I - \tau_l L_{\mathrm{g}})B^{(l)}, \qquad \tau_l = \tanh(\theta_l) \in (-1, 1). \end{aligned} \tag{5}$$

Here θ_l is trainable. The first line defines DPC propagation; adding SSC gives DPC-GCN-S. In the backbone, SSC acts after linear additions and before normalization (Appendix C.3).

3.1 DEGREE-POTENTIAL CALIBRATION

The receiver-dependent degree-potential contrasts in Figure 1(a) motivate a topology coordinate that adapts the shared calibration to each node's neighborhood.

Centered topology coordinate. For receiving node i, define its squared edge-weight discrepancy and bounded relative score as

$$e_i = \|\Delta_{i:}\|_2^2 = \sum_j (P_{\mathrm{g}})_{ij}^2 \kappa_{ij}^2, \qquad q_i = \frac{\|\Delta_{i:}\|_2}{\|(P_{\mathrm{g}})_{i:}\|_2 + \|\Delta_{i:}\|_2 + \epsilon}. \tag{6}$$

The bounded score $q_i \in [0, 1)$ measures relative propagation discrepancy, while Δ retains the signed edge corrections; $\epsilon > 0$ provides numerical protection. For small contrasts, $\kappa_{ij} \approx \phi_j - \phi_i$, so $\sqrt{e_i}$ measures weighted local degree-potential variation; Appendix A.1 bounds the finite-contrast case. We remove the global part of $\mathrm{Diag}(q)\Delta$ by centering with weights e_i:

$$\mu_\Delta = \frac{\sum_i e_i q_i}{\sum_i e_i}, \qquad t_i = \frac{q_i - \mu_\Delta}{\max_k |q_k - \mu_\Delta|}. \tag{7}$$

Set $\mu_\Delta = 0$ and $t = 0$ when $\sum_i e_i = 0$, and $t = 0$ when the denominator vanishes. Centering removes the best global Frobenius fit of $\mathrm{Diag}(q)\Delta$ (Appendix A.2); normalization bounds $|t_i| \le 1$, so λ_l controls the node-wise correction scale. The sign of t_i locates the receiver above or below the weighted center.

Operator formulation. The coefficient $\alpha_l \in [0, 1]$ interpolates between the reference operators, while $\lambda_l \in (-1, 1)$ controls signed node-wise deviations. Using $\Delta_{ij} = (P_{\mathrm{g}})_{ij}\kappa_{ij}$, Equation (5) induces

$$P_{\mathrm{DPC}}^{(l)} = P_{\mathrm{g}} + \mathrm{Diag}(c^{(l)})\Delta = P_{\mathrm{g}} + \alpha_l \Delta + \lambda_l T, \qquad T = \mathrm{Diag}(t)\Delta. \tag{8}$$

The choices $(\alpha_l, \lambda_l) = (0, 0)$ and $(1, 0)$ recover symmetric GCN and row-normalized mean propagation, respectively. When adjacent nodes have equal effective degrees, the contrasts and corrections vanish, recovering the common reference propagation (Appendix A.1).

Proposition 1 (Node-wise correction beyond global mixing)**.** For $\alpha_l \in [0, 1]$, the squared Frobenius distance from the DPC operator to the global mixing family is

$$\min_{\beta \in [0,1]} \|P_{\mathrm{DPC}}^{(l)} - (P_{\mathrm{g}} + \beta\Delta)\|_F^2 = \lambda_l^2 \|T\|_F^2. \tag{9}$$

Thus a nonzero correction $\lambda_l T$ cannot be replaced by adjusting the global coefficient alone. The residual direction reflects variation in q_i among receivers with nonzero discrepancy. Moreover, $\|P_{\mathrm{DPC}}^{(l)} - P_{\mathrm{g}}\|_F \le \sqrt{\alpha_l^2 + \lambda_l^2}\|\Delta\|_F$, so small degree-potential discrepancy limits departure from the reference propagation. Appendix A.2 proves the proposition and these consequences.

3.2 SIGNED SPECTRAL CALIBRATION

Signed state calibration. A first-order correction with the fixed L_{g} preserves the state at zero strength and uses one extra sparse propagation. We use it to adjust the frequency response of the state produced by DPC. Consider first the more general node-wise scaling:

$$\widehat{B}^{(l)} = (I - D_l L_{\mathrm{g}})B^{(l)}, \qquad D_l = \mathrm{Diag}(\eta_{l,1}, \ldots, \eta_{l,N}), \tag{10}$$

Here $\eta_{l,i}$ scales the correction at node i. To adjust frequencies in a common reference basis, we require a scalar spectral response $f(L_{\mathrm{g}})$ (Shuman et al., 2013).

Lemma 1 (Shared spectral coefficients)**.** *On a connected undirected graph, a real diagonal D_l satisfies $I - D_l L_{\mathrm{g}} = f(L_{\mathrm{g}})$ for a real-valued spectral function f if and only if $D_l = \tau_l I$ for some scalar τ_l.*

The lemma motivates sharing τ_l across nodes in Equation (5) to preserve a scalar spectral response (proof in Appendix A.3). Sharing across features is an additional compact parameterization choice; its empirical effect is assessed in Section 4.4. We learn the bounded coefficient through classification supervision.

Spectral and energy response.

Proposition 2 (One-step signed spectral-energy response)**.** For an SSC step with input state B and output $\widehat{B} = (I - \tau L_{\mathrm{g}})B$, a graph Fourier mode of frequency μ has amplitude response $g_\tau(\mu) = 1 - \tau\mu$, and

$$\mathcal{E}(\widehat{B}) - \mathcal{E}(B) = -\tau \|L_{\mathrm{g}} B\|_F^2 + \frac{\tau^2}{2} \operatorname{Tr}(B^\top L_{\mathrm{g}}^3 B). \tag{11}$$

If $0 < \tau < 1$, every active nonzero-frequency component strictly loses energy; if $-1 < \tau < 0$, every such component strictly gains energy; and $\tau = 0$ is the identity.

The learned sign therefore selects smoothing or sharpening of the aggregated state. For $-1 < \tau < 0$, $g_\tau(\mu)$ increases with frequency and satisfies $1 \le g_\tau(\mu) < 3$, providing preferential high-frequency amplification with a bounded single-step amplitude gain. Appendix A.4 proves these properties.

Why learn the direction? For a fixed binary classifier, consider node scores $z \sim \mathcal{N}(m, \Sigma)$ with $\Sigma_{ii} > 0$, labels $y_i \in \{-1, +1\}$, and calibrated scores $z(\tau) = (I - \tau L_{\mathrm{g}})z$. Define the standardized margin $\gamma_i(\tau) = y_i \mathbb{E}[z_i(\tau)] / \sqrt{\mathrm{Var}(z_i(\tau))}$. For predictions $\mathrm{sign}(z(\tau))$, the expected misclassification rate $R(\tau)$ satisfies

$$R'(0) = -\frac{1}{N} \sum_i \varphi(\gamma_i(0)) \gamma_i'(0), \tag{12}$$

where φ is the standard Gaussian density. Thus the beneficial local direction depends on changes in node margins, weighted most strongly near the classification boundary. We next examine what label spectral energy determines on regular graphs, where DPC reduces to P_{g}. Let Π_μ project onto the frequency-μ eigenspace and define $p_y(\mu) = \|\Pi_\mu y\|_2^2$.

Proposition 3 (Spectral energy does not determine the beneficial direction)**.** Consider binary labels $y \in \{-1, +1\}^N$, features $x = y + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, and predictions $\mathrm{sign}((I - \tau L_{\mathrm{g}})P_{\mathrm{g}} x)$. Let $R_y(\tau)$ be the expected fraction of misclassified nodes. There exist arbitrarily large connected regular graphs and two balanced labelings $y^{\mathrm{A}}, y^{\mathrm{B}}$ on the same graph with identical $p_y(\mu)$ at every frequency, identical initial standardized-margin distributions, and identical $R_y(0)$, yet $R'_{y^{\mathrm{A}}}(0) < 0 < R'_{y^{\mathrm{B}}}(0)$ for a nonempty open interval of $\sigma > 0$.

Even with these quantities fixed, which nodes benefit from calibration can change the favorable direction. The same construction shows that lower reconstruction error and Dirichlet energy can accompany higher classification risk (Appendix A.5). These results motivate learning SSC's sign from classification supervision, allowing the task to select smoothing or sharpening.

Computational cost. Topology preprocessing and storage, and per-layer edge-weight construction, each cost $\mathcal{O}(N + |E|)$. DPC uses one sparse matrix multiplication (SpMM) per layer; SSC adds one with P_{g}. A depth-L network adds $2L$ or $3L$ trainable scalars, respectively. Both retain sparse GCN's layer complexity $\mathcal{O}(N d_{l-1} d_l + |E| d_l + N d_l)$ for input and output widths d_{l-1} and d_l.

4 Experiments

Our experiments assess whether calibrated local graph convolution remains competitive with tuned GNNs and graph Transformers across graph types and scales, how DPC and SSC contribute, and whether alternative calibration designs offer consistent advantages. We further examine learned calibration behavior, initialization sensitivity, and training-time and memory costs. Sections 4.2–4.5 present the main performance, ablation, design, and efficiency comparisons; Appendix B provides supplementary diagnostics and analyses, while Appendix C details the evaluation and model configurations.

4.1 Experimental Setup

Datasets. We evaluate eight homophilous graphs (Cora, CiteSeer, PubMed, Amazon-Computer, Amazon-Photo, Coauthor-CS, Coauthor-Physics, WikiCS), five heterophilous graphs (Amazon-Ratings, Minesweeper, Roman-Empire, Questions, Tolokers) (Platonov et al., 2023), and three large-scale graphs: ogbn-arxiv, ogbn-products (Hu et al., 2020), and Pokec (Lim et al., 2021). We

report ROC-AUC on Minesweeper, Questions, and Tolokers, and accuracy elsewhere. Results are reported as mean test performance ± standard deviation over five runs on medium-scale graphs and three on large-scale graphs, using the best-validation checkpoint of each run. Appendices C.1–C.4 provide evaluation and model settings.

Baselines. We compare with dataset-specifically tuned GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2018), denoted Tuned-GCN, Tuned-GraphSAGE, and Tuned-GAT, following the Classic GNN study (Luo et al., 2024). Co-GNN (Finkelshtein et al., 2024) provides a node-adaptive communication comparison, while AdaGNN-S (Dong et al., 2021) and CPF (Li et al., 2025) provide channel-wise and partition-wise adaptive-filtering comparisons, respectively. We also include H2GCN (Zhu et al., 2020), Polynormer (Deng et al., 2024), GAMMA (Ghazizadeh et al., 2025), GT-SNT (Zhang et al., 2026), DUALFormer (Zhuo et al., 2025), and SGFormer (Wu et al., 2023b).

4.2 Medium-Scale Results

Table 1: Node classification results on homophilous graphs (%). We highlight the top first, second, and third results per dataset.

	Cora	CiteSeer	PubMed	Computer	Photo	CS	Physics	WikiCS
Metric	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑
SGFormer	82.68±0.76	68.56±0.66	78.36±1.01	91.01±1.53	95.49±0.11	94.64±0.43	96.55±0.12	79.22±0.62
Polynormer	75.42±0.74	64.74±1.15	75.42±0.86	93.71±0.35	96.50±0.29	95.46±0.22	97.11±0.07	79.83±0.94
DUALFormer	83.84±0.52	71.54±0.82	80.62±0.43	92.54±0.16	95.02±0.18	95.12±0.16	97.06±0.04	80.45±0.50
GT-SNT	84.76±0.57	71.18±1.28	79.94±1.39	92.84±0.11	95.16±0.18	94.89±0.14	97.23±0.03	79.48±0.76
AdaGNN-S	84.06±0.57	71.88±0.76	80.34±0.48	86.11±0.11	94.18±0.49	95.83±0.04	97.24±0.03	79.56±0.69
H2GCN	83.88±0.54	71.46±0.47	80.70±0.45	91.20±0.19	95.88±0.16	95.93±0.12	97.13±0.02	79.17±0.62
Tuned-GCN	84.54±0.86	72.64±0.40	80.70±0.96	93.86±0.08	95.93±0.31	95.64±0.23	97.47±0.07	80.03±0.55
Tuned-GraphSAGE	83.52±0.61	71.64±0.47	77.94±0.96	93.05±0.23	96.18±0.18	96.06±0.20	97.22±0.07	80.54±0.41
Tuned-GAT	82.40±1.01	71.86±0.35	79.76±1.21	93.73±0.16	96.50±0.34	96.16±0.08	97.13±0.16	80.66±0.75
GAMMA	81.18±0.58	66.58±1.30	78.40±0.41	92.02±0.12	96.03±0.22	95.28±0.13	96.93±0.05	79.76±0.42
CPF	81.64±0.87	69.24±1.63	79.52±0.35	92.60±0.21	96.22±0.16	95.95±0.09	97.23±0.02	80.17±0.57
DPC-GCN	**85.22±0.90**	**73.28±0.60**	**81.36±0.51**	**94.20±0.10**	**96.54±0.12**	**96.26±0.10**	**97.47±0.05**	**80.90±0.28**
DPC-GCN-S	**85.26±0.64**	**73.08±0.54**	**81.42±0.39**	**94.21±0.22**	**96.51±0.13**	**96.25±0.11**	**97.46±0.05**	**80.83±0.25**

Table 2: Node classification results on heterophilous graphs (%). We highlight the top first, second, and third results per dataset. OOM means out of memory.

	Amazon-Ratings	Minesweeper	Roman-Empire	Questions	Tolokers
Metric	Accuracy↑	ROC-AUC↑	Accuracy↑	ROC-AUC↑	ROC-AUC↑
SGFormer	49.76±0.33	78.08±0.99	73.24±0.41	78.19±0.79	83.18±0.62
Polynormer	54.46±0.23	97.09±0.32	92.51±0.32	78.15±1.43	OOM
DUALFormer	48.01±0.66	82.20±0.76	46.04±0.74	62.63±1.42	75.53±1.24
GT-SNT	49.30±0.34	87.83±0.53	75.26±0.58	77.26±1.02	82.02±0.74
AdaGNN-S	45.59±1.37	91.05±0.28	79.58±0.25	76.70±0.53	83.91±0.99
H2GCN	47.23±0.42	90.73±0.10	78.59±0.46	74.87±1.27	82.68±0.72
Co-GNN	53.84±0.49	95.93±0.38	90.10±0.35	78.64±1.37	84.00±0.96
Tuned-GCN	53.75±0.47	97.27±0.35	91.13±0.21	78.62±0.66	86.01±0.54
Tuned-GraphSAGE	55.24±0.66	97.13±0.89	90.47±0.32	76.46±0.90	85.23±0.42
Tuned-GAT	55.31±0.64	97.26±0.87	90.07±0.55	77.54±0.71	85.78±0.31
GAMMA	45.31±0.48	89.81±0.90	74.00±0.28	75.52±0.92	80.37±0.90
CPF	49.84±0.50	77.58±0.99	75.23±0.59	74.37±1.15	78.79±0.96
DPC-GCN	**55.42±0.38**	**97.73±0.38**	**91.58±0.22**	**79.06±0.54**	**86.41±0.56**
DPC-GCN-S	**55.55±0.39**	**98.73±0.19**	**91.97±0.22**	**79.89±0.63**	**86.62±0.53**

Homophilous graphs. DPC-GCN matches or exceeds the best baseline mean on all eight homophilous datasets (Table 1). It improves over the strongest baseline by 0.64 points on CiteSeer and

0.66 on PubMed, and ties Tuned-GCN on Physics. These results establish the competitiveness of degree-calibrated local propagation against both tuned GNNs and graph Transformers.

Heterophilous graphs. DPC-GCN-S achieves the highest mean on four of the five heterophilous datasets (Table 2), exceeding the strongest baseline by 1.46 points on Minesweeper and 1.25 on Questions. Polynormer leads on Roman-Empire by 0.54 points. Section 4.4 examines the component contributions.

4.3 Large-Scale Results

To assess applicability to larger graphs, we use full-graph training on ogbn-arxiv and partitioned or mini-batch training on ogbn-products and Pokec (Appendix C.3). In Table 3, DPC-GCN reaches 73.40% on ogbn-arxiv, close to Polynormer's 73.39%, while DPC-GCN-S achieves the highest mean on Pokec, exceeding Tuned-GCN by 0.23 points. On ogbn-products, DPC-GCN trails Polynormer by 2.27 points but uses less time per epoch and less peak memory (Appendix B.6).

Table 3: Node classification results on large-scale graphs (%). We highlight the top first, second, and third results per dataset.

	ogbn-arxiv	ogbn-products	Pokec
Metric	Accuracy↑	Accuracy↑	Accuracy↑
Polynormer	73.39±0.30	83.59±0.09	86.20±0.08
DUALFormer	66.21±0.12	76.27±0.05	56.13±0.03
GT-SNT	71.80±0.24	71.10±0.23	79.42±0.25
Tuned-GCN	73.05±0.16	81.12±0.33	86.24±0.06
Tuned-GraphSAGE	72.76±0.23	82.77±0.35	85.95±0.09
Tuned-GAT	72.38±0.04	82.08±0.52	86.01±0.08
GAMMA	70.00±0.08	72.26±0.17	77.75±0.19
DPC-GCN	**73.40±0.11**	81.32±0.12	**86.25±0.06**
DPC-GCN-S	72.78±0.19	81.26±0.25	**86.47±0.06**

4.4 Ablation and Design Analysis

Component contributions. We assess DPC and SSC across graph types and the two corrections within DPC under matched settings (Tables 4–5). The last two rows remove the shared ($\alpha_l \equiv 0$) or receiver-dependent ($\lambda_l \equiv 0$) correction from DPC-GCN, with SSC disabled.

On homophilous graphs, DPC benefits several tasks, whereas SSC has limited additional effect (Table 4). With SSC retained, DPC adds 0.92 points on Photo and 0.54 on WikiCS. Within DPC-GCN, removing the shared correction lowers CiteSeer by 0.72 points; removing the local correction lowers Cora by 0.64. SSC changes the mean by -0.20 to $+0.06$ points, accompanied by smaller learned coefficient magnitudes than on heterophilous graphs (Appendix B.4).

Table 4: Component ablations on homophilous graphs (accuracy in %, five-run mean ± standard deviation). Bold marks highest displayed means, including ties.

Variant	Cora	CiteSeer	PubMed	Computer	Photo	CS	Physics	WikiCS
DPC-GCN-S	**85.26±0.64**	73.08±0.54	81.42±0.39	**94.21±0.22**	96.51±0.13	96.25±0.11	97.46±0.05	80.83±0.25
w/o SSC	85.22±0.90	**73.28±0.60**	81.36±0.51	94.20±0.10	96.54±0.12	**96.26±0.10**	**97.47±0.05**	**80.90±0.28**
w/o DPC	84.98±0.44	72.76±0.62	**81.58±0.33**	93.86±0.19	95.59±0.21	96.15±0.10	97.43±0.04	80.29±0.23
w/o SSC, $\alpha_l \equiv 0$	84.70±0.41	72.56±0.36	80.84±0.50	94.10±0.21	95.93±0.42	96.15±0.10	97.46±0.05	80.20±0.38
w/o SSC, $\lambda_l \equiv 0$	84.58±0.60	73.04±0.74	80.88±0.81	94.01±0.17	**96.55±0.16**	**96.26±0.10**	97.42±0.05	80.71±0.37

On heterophilous graphs, SSC improves the mean on all five datasets (Table 5), with gains of 1.00 points on Minesweeper and 0.83 on Questions. With SSC retained, DPC contributes most on Amazon-Ratings (+1.20 points). Within DPC-GCN, removing the shared correction reduces Amazon-Ratings performance by 1.20 points, while removing the receiver-dependent correction reduces Minesweeper performance by 0.41 points. Thus, SSC provides more consistent gains on these tasks, while the contributions of DPC and its two correction terms depend on the dataset.

Table 5: Component ablations on heterophilous graphs (%, five-run mean ± standard deviation). Ratings and Roman use accuracy; others use ROC-AUC. Bold marks highest displayed means, including ties.

Variant	Ratings	Minesweeper	Roman	Questions	Tolokers
DPC-GCN-S	**55.55±0.39**	**98.73±0.19**	**91.97±0.22**	**79.89±0.63**	**86.62±0.53**
w/o SSC	55.42±0.38	97.73±0.38	91.58±0.22	79.06±0.54	86.41±0.56
w/o DPC	54.35±0.16	98.58±0.32	**91.97±0.16**	**79.89±0.63**	86.52±0.57
w/o SSC, $\alpha_l \equiv 0$	54.22±0.29	97.91±0.25	91.66±0.15	79.05±0.55	86.46±0.52
w/o SSC, $\lambda_l \equiv 0$	55.28±0.17	97.32±0.39	91.59±0.25	79.08±0.52	86.37±0.51

Design choices. Coordinate controls favor centered DPC on Computer, WikiCS, and Minesweeper. For SSC, the relative performance of smooth-only and sharp-only restrictions varies across tasks, supporting a learned sign. Finer coefficients give no consistent gain, while fixed extra propagation performs substantially worse. These comparisons favor a shared scalar with learned direction and strength (Appendices B.2 and B.3).

4.5 Efficiency

Figure 3 compares predictive performance, per-epoch training time, and peak memory at each method's selected configuration. On WikiCS, DPC-GCN improves accuracy over Tuned-GCN by 0.87 points while reducing epoch time to approximately one third and peak memory by 88.8%. On Minesweeper, DPC-GCN-S improves ROC-AUC over Polynormer by 1.64 points with $1.76\times$ faster epochs and 93.5% lower peak memory. Together, these comparisons show that our methods can improve predictive performance while reducing both training time per epoch and memory usage. Appendix C.2 provides hardware details.

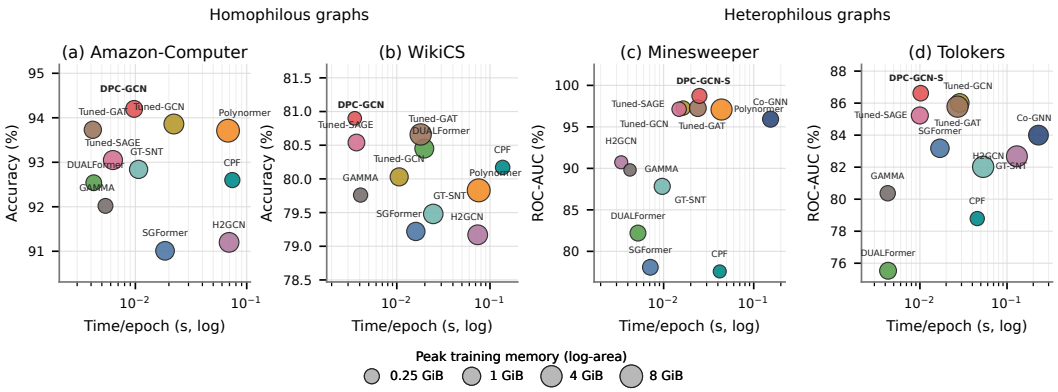

Figure 3: Performance versus per-epoch training time, with circle area encoding log-transformed peak training memory. Panels (a)–(b) show DPC-GCN on homophilous datasets, and panels (c)–(d) show DPC-GCN-S on heterophilous datasets.

5 Conclusion

We presented DPC-GCN and its signed spectral extension, DPC-GCN-S, for local graph convolution. DPC combines shared and receiver-dependent corrections based on source–receiver degree relations, while SSC learns smoothing or sharpening with one scalar per layer. Experiments on 16 benchmarks demonstrate competitive performance against tuned GNNs and graph Transformers, with spectral calibration benefiting the tested heterophilous tasks more consistently. These findings highlight the value of calibrating both message weights and aggregated states within a sparse local architecture. Evaluation beyond transductive classification on static undirected graphs, including broader architecture transfer and inductive or dynamic settings, remains future work.

AI USE STATEMENT

We selected the literature, determined the theoretical claims, designed the main experiments and evaluation settings, and provided the initial drafts and argument structure. Generative AI tools assisted with literature search, completing and checking mathematical derivations and proofs, writing and debugging experimental and result-aggregation scripts, and revising our drafts and polishing the language. They also offered suggestions for selected supplementary analyses and for explaining and presenting experimental results. We reviewed the AI-assisted work, made the final decisions, and take responsibility for the accuracy and validity of the manuscript, including its claims, proofs, code, citations, and results.

REPRODUCIBILITY STATEMENT

The method definition, assumptions, and theoretical guarantees are provided in Sections 2 and 3, with complete proofs in Appendix A. Dataset processing, split construction, evaluation protocols, hyperparameters, and baseline configurations are documented in Appendix C. The implementation will be available through the accompanying code link.

REFERENCES

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, 2022. doi: 10.1109/TPAMI.2021.3054830.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*, 2014.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 18–24 Jul 2021a.

Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami flow and neural diffusion on graphs. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1594–1609. Curran Associates, Inc., 2021b.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13260–13271. Curran Associates, Inc., 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 18323–18348, 2024.

Yushun Dong, Kaize Ding, Brian Jalaian, Shuiwang Ji, and Jundong Li. Adagnn: Graph neural networks with adaptive frequency response filter. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 392–401, 2021.

Moshe Eliasof, Eldad Haber, and Eran Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3836–3849. Curran Associates, Inc., 2021.

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

Ben Finkelshtein, Xingyue Huang, Michael M. Bronstein, and Ismail Ilkan Ceylan. Cooperative graph neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13633–13659. PMLR, 21–27 Jul 2024.

Amir Ghazizadeh, Rickard Ewetz, and Hao Zheng. Gamma: Gated multi-hop message passing for homophily-agnostic node representation in gnns. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 11726–11777. Curran Associates, Inc., 2025. doi: 10.52202/085713-0393.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017.

Ming Gu, Zhuonan Zheng, Sheng Zhou, Meihan Liu, Jiawei Chen, Qiaoyu Tan, Liangcheng Li, and Jiajun Bu. Making classic gnns strong baselines across varying homophily: A smoothness–generalization perspective. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 51025–51060. Curran Associates, Inc., 2025. doi: 10.52202/085713-1701.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Mingguo He, Zhewei Wei, zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 14239–14251. Curran Associates, Inc., 2021.

Van Thuy Hoang, Hyeon-Ju Jeon, and O-Joun Lee. Mitigating degree bias in graph representation learning with learnable structural augmentation and structural self-attention. *IEEE Transactions on Network Science and Engineering*, 12(5):3656–3670, 2025. doi: 10.1109/TNSE.2025.3563697.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133. Curran Associates, Inc., 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C. Bayan Bruss, and Tom Goldstein. GOAT: A global transformer on large-scale graphs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17375–17390. PMLR, 23–29 Jul 2023.

Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21618–21629. Curran Associates, Inc., 2021.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2019. doi: 10.1109/TSP.2018.2879624.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

Guoming Li, Jian Yang, and Yifan Chen. Partition-wise graph filtering: A unified perspective through the lens of graph coarsening. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1353–1364, 2025.

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. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20887–20902. Curran Associates, Inc., 2021.

Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. On generalized degree fairness in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4525–4533, 2023.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1362–1375. Curran Associates, Inc., 2022. doi: 10.52202/068431-0100.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic gnns are strong baselines: Reassessing gnns for node classification. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97650–97669. Curran Associates, Inc., 2024. doi: 10.52202/079017-3098.

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 *The Eleventh International Conference on Learning Representations*, 2023.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515. Curran Associates, Inc., 2022. doi: 10.52202/068431-1054.

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. doi: 10.1109/MSP.2012.2235192.

Arjun Subramonian, Jian Kang, and Yizhou Sun. Theoretical and empirical insights into the origins of degree bias in graph neural networks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 8193–8239. Curran Associates, Inc., 2024. doi: 10.52202/079017-0264.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23341–23362. PMLR, 17–23 Jul 2022.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27387–27401. Curran Associates, Inc., 2022. doi: 10.52202/068431-1986.

Qitian Wu, Chenxiao Yang, Wentao Zhao, Yixuan He, David Wipf, and Junchi Yan. DIFFormer: Scalable (graph) transformers induced by energy constrained diffusion. In *International Conference on Learning Representations (ICLR)*, 2023a.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64753–64773. Curran Associates, Inc., 2023b. doi: 10.52202/075280-2826.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888. Curran Associates, Inc., 2021.

Huizhe Zhang, Jintang Li, Yuchang Zhu, Huazhen Zhong, and Liang Chen. Gt-snt: a linear-time transformer for large-scale graphs via spiking node tokenization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16298–16306, 2026.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7793–7804. Curran Associates, Inc., 2020.

Jiaming Zhuo, Yuwei Liu, Yintong Lu, Ziyi Ma, Kun Fu, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. DUALFormer: Dual graph transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Theoretical Analysis

Appendix A.1 relates the propagation discrepancy to degree-potential contrasts. Appendix A.2 proves the DPC decomposition and its weighted-variance consequence. Appendix A.3 proves the shared-coefficient condition, and Appendix A.4 derives the SSC response. Appendix A.5 derives the margin criterion, constructs tasks with matched spectra and initial margins but opposite beneficial directions, and proves the reconstruction consequence.

A.1 Degree-Potential Representation

For an edge $j \to i$, let $\delta_{ij} = \phi_j - \phi_i$. Substitution of Equation (1) gives $(P_{\mathrm{m}})_{ij}/(P_{\mathrm{g}})_{ij} = \sqrt{\widetilde{d}_j/\widetilde{d}_i} = e^{\delta_{ij}}$ on the graph support. Hence $\Delta_{ij} = (P_{\mathrm{g}})_{ij}(e^{\delta_{ij}} - 1)$, which is the exact degree-potential representation in Equation (4).

To quantify its local interpretation, define

$$\rho_i = \max_{j:\widetilde{A}_{ij}>0} |\delta_{ij}|, \qquad a_i = \left(\sum_j (P_{\mathrm{g}})_{ij}^2 \delta_{ij}^2 \right)^{1/2}.$$

The mean-value theorem gives $e^{\delta_{ij}} - 1 = e^{\xi}\delta_{ij}$ for some ξ between 0 and δ_{ij}. Hence, on each incident edge,

$$e^{-\rho_i}|\delta_{ij}| \le |e^{\delta_{ij}} - 1| \le e^{\rho_i}|\delta_{ij}|.$$

Squaring, weighting by $(P_{\mathrm{g}})_{ij}^2$, and summing over j gives $e^{-2\rho_i}a_i^2 \le e_i \le e^{2\rho_i}a_i^2$. Taking square roots yields

$$e^{-\rho_i}a_i \le \sqrt{e_i} \le e^{\rho_i}a_i. \tag{13}$$

Thus $\sqrt{e_i}$ measures weighted local degree-potential variation with multiplicative distortion controlled by the largest incident contrast. As $\rho_i \to 0$, the two bounds approach a_i, recovering the first-order interpretation used in Section 3.1.

A.2 DPC Decomposition

Proof of Proposition 1. Let $V_q = \mathrm{Diag}(q)\Delta$ and $E_\Delta = \|\Delta\|_F^2 = \sum_i e_i$. When $E_\Delta > 0$, the best global approximation of the raw node-wise correction is

$$\arg\min_{b\in\mathbb{R}} \|V_q - b\Delta\|_F^2 = \frac{\langle \Delta, V_q\rangle_F}{E_\Delta} = \frac{\sum_i e_i q_i}{\sum_i e_i} = \mu_\Delta. \tag{14}$$

Indeed, differentiating $\sum_i e_i(q_i - b)^2$ gives this unique minimizer. For $M = \max_i |q_i - \mu_\Delta| > 0$, the residual is $V_q - \mu_\Delta\Delta = MT$. The centering therefore yields

$$\langle \Delta, T\rangle_F = \sum_i e_i t_i = \frac{\sum_i e_i(q_i - \mu_\Delta)}{M} = 0.$$

If $E_\Delta = 0$ or $M = 0$, the convention $T = 0$ gives the same orthogonality. For every $\beta \in [0,1]$, expanding the squared distance now gives

$$\|P_{\mathrm{DPC}}^{(l)} - (P_{\mathrm{g}} + \beta\Delta)\|_F^2 = (\alpha_l - \beta)^2\|\Delta\|_F^2 + \lambda_l^2\|T\|_F^2.$$

Because $\alpha_l \in [0,1]$, the minimum is attained at $\beta = \alpha_l$, proving Equation (9). □

Consequences of the decomposition. For $E_\Delta > 0$ and $M > 0$, let $w_i = e_i/E_\Delta$. Substituting $t_i = (q_i - \mu_\Delta)/M$ into the residual norm gives

$$\begin{aligned} \|T\|_F^2 &= \sum_i e_i t_i^2 = \frac{E_\Delta}{M^2}\sum_i w_i(q_i - \mu_\Delta)^2 \\ &= \frac{E_\Delta}{M^2}\,\mathrm{Var}_w(q). \end{aligned} \tag{15}$$

A weighted variance is zero exactly when its values agree on all indices with positive weight. Consequently, $T \neq 0$ exactly when q_i varies among receivers with $e_i > 0$.

Thus nonuniform relative discrepancy supplies a node-wise direction beyond global mixing; its actual contribution also requires $\lambda_l \neq 0$. The decomposition additionally controls the size of the total correction. Orthogonality and $|t_i| \leq 1$ give

$$\begin{aligned} \|P_{\mathrm{DPC}}^{(l)} - P_{\mathrm{g}}\|_F^2 &= \alpha_l^2 \|\Delta\|_F^2 + \lambda_l^2 \|T\|_F^2 \\ &\leq (\alpha_l^2 + \lambda_l^2)\|\Delta\|_F^2 \leq 2\|\Delta\|_F^2. \end{aligned} \tag{16}$$

This proves the perturbation bound stated after Proposition 1.

A.3 Shared Spectral Coefficients

Proof of Lemma 1. For clarity, omit the layer index and write $D = \mathrm{Diag}(\eta_1, \ldots, \eta_N)$. The coefficients η_i are initially unrestricted real numbers and need not agree. We prove necessity and sufficiency separately.

Necessity. Suppose $I - DL_{\mathrm{g}} = f(L_{\mathrm{g}})$. Since L_{g} is real symmetric, it has an orthogonal eigendecomposition $L_{\mathrm{g}} = U\Lambda U^\top$. The standard definition of a real-valued spectral filter (Shuman et al., 2013) gives

$$f(L_{\mathrm{g}}) = U f(\Lambda) U^\top, \qquad f(\Lambda) = \mathrm{Diag}(f(\mu_1), \ldots, f(\mu_N)).$$

Because $f(\Lambda)$ is real diagonal, $f(L_{\mathrm{g}})^\top = f(L_{\mathrm{g}})$. The assumed update must therefore also be symmetric. Using $D^\top = D$ and $L_{\mathrm{g}}^\top = L_{\mathrm{g}}$,

$$\begin{aligned} I - DL_{\mathrm{g}} &= f(L_{\mathrm{g}}) = f(L_{\mathrm{g}})^\top \\ &= (I - DL_{\mathrm{g}})^\top = I - L_{\mathrm{g}} D, \end{aligned} \tag{17}$$

so $DL_{\mathrm{g}} - L_{\mathrm{g}} D = 0$. To see what this requires of the coefficients, expand the two matrix products entrywise. Diagonality gives

$$\begin{aligned} (DL_{\mathrm{g}})_{ij} &= \sum_k D_{ik} (L_{\mathrm{g}})_{kj} = \eta_i (L_{\mathrm{g}})_{ij}, \\ (L_{\mathrm{g}} D)_{ij} &= \sum_k (L_{\mathrm{g}})_{ik} D_{kj} = (L_{\mathrm{g}})_{ij} \eta_j. \end{aligned}$$

Subtracting yields

$$(\eta_i - \eta_j)(L_{\mathrm{g}})_{ij} = 0 \quad \text{for all } i, j. \tag{18}$$

For adjacent distinct nodes i, j, the normalized Laplacian satisfies

$$(L_{\mathrm{g}})_{ij} = -\frac{\widetilde{A}_{ij}}{\sqrt{\widetilde{d}_i \widetilde{d}_j}} \neq 0.$$

Equation (18) thus implies $\eta_i = \eta_j$ on every edge. For any two nodes u, v, connectivity supplies a path $u = i_0, i_1, \ldots, i_m = v$; applying this equality along the path gives $\eta_u = \eta_{i_1} = \cdots = \eta_v$. All diagonal entries therefore equal a common scalar τ, so $D = \tau I$.

Sufficiency. Conversely, if $D = \tau I$, then

$$I - DL_{\mathrm{g}} = I - \tau L_{\mathrm{g}} = U(I - \tau\Lambda)U^\top = f(L_{\mathrm{g}}), \qquad f(\mu) = 1 - \tau\mu.$$

This is a real-valued scalar spectral response, completing both directions. □

Sharing across nodes preserves the scalar spectral response. Separate coefficients for individual features also preserve a scalar spectral response within each channel.

A.4 SSC Response

Proof of Proposition 2. Write $L_{\mathrm{g}} = U\,\mathrm{Diag}(\mu_1, \ldots, \mu_N)U^\top$, where U is orthogonal and $0 \le \mu_k \le 2$ for the symmetric normalized Laplacian (Chung, 1997; Shuman et al., 2013), and let $\Gamma = U^\top B$. Then

$$\mathcal{E}(B) = \frac{1}{2}\sum_{k=1}^{N} \mu_k \|\Gamma_{k:}\|_2^2, \qquad U^\top \widehat{B} = \mathrm{Diag}(1-\tau\mu_k)\Gamma.$$

The second identity gives the amplitude response $g_\tau(\mu) = 1 - \tau\mu$. Subtracting the input energy from the output energy yields

$$\begin{aligned}
\mathcal{E}(\widehat{B}) - \mathcal{E}(B) &= \frac{1}{2}\sum_k \mu_k \left[(1-\tau\mu_k)^2 - 1\right] \|\Gamma_{k:}\|_2^2 \\
&= -\tau \sum_k \mu_k^2 \|\Gamma_{k:}\|_2^2 + \frac{\tau^2}{2}\sum_k \mu_k^3 \|\Gamma_{k:}\|_2^2 \\
&= -\tau \|L_{\mathrm{g}} B\|_F^2 + \frac{\tau^2}{2}\,\mathrm{Tr}(B^\top L_{\mathrm{g}}^3 B).
\end{aligned}$$

This proves Equation (11). For $0 < \tau < 1$ and $\mu_k > 0$, $|1 - \tau\mu_k| < 1$, so every active nonzero-frequency component strictly loses energy. For $-1 < \tau < 0$, $1 - \tau\mu_k > 1$, so each such component strictly gains energy. At $\tau = 0$, the update is the identity. Finally, writing $\tau = -s$ with $0 < s < 1$ gives $g_{-s}(\mu) = 1 + s\mu$, which increases with frequency and satisfies $1 \le g_{-s}(\mu) \le 1 + 2s < 3$. □

A.5 Classification Direction

Derivation of Equation (12) *and proof of Proposition 3.* **Step 1: general margin criterion.** Write $T_\tau = I - \tau L_{\mathrm{g}}$. Under the Gaussian score model in the main text, $\Sigma \succeq 0$ and $\Sigma_{ii} > 0$. For τ near zero, the marginal standard deviation and standardized margin are

$$s_i(\tau) = \sqrt{(T_\tau \Sigma T_\tau^\top)_{ii}}, \qquad \gamma_i(\tau) = \frac{y_i (T_\tau m)_i}{s_i(\tau)}. \tag{19}$$

Gaussian marginals give $R(\tau) = N^{-1}\sum_i \Phi(-\gamma_i(\tau))$, where Φ is the standard Gaussian distribution function. To differentiate the margins, expand their variances:

$$T_\tau \Sigma T_\tau^\top = \Sigma - \tau(L_{\mathrm{g}}\Sigma + \Sigma L_{\mathrm{g}}) + \tau^2 L_{\mathrm{g}}\Sigma L_{\mathrm{g}}.$$

Symmetry implies $(L_{\mathrm{g}}\Sigma)_{ii} = (\Sigma L_{\mathrm{g}})_{ii}$, so $s_i'(0)/s_i(0) = -(L_{\mathrm{g}}\Sigma)_{ii}/\Sigma_{ii}$. Differentiating the numerator and denominator in Equation (19) yields

$$v_i := \gamma_i'(0) = -\frac{y_i (L_{\mathrm{g}} m)_i}{\sqrt{\Sigma_{ii}}} + \gamma_i(0)\frac{(L_{\mathrm{g}}\Sigma)_{ii}}{\Sigma_{ii}}. \tag{20}$$

Differentiating R and using $\Phi' = \varphi$ gives Equation (12). Since $R(\tau) - R(0) = R'(0)\tau + o(|\tau|)$, a nonzero derivative determines the risk-reducing sign for sufficiently small calibration.

Step 2: matched spectra and initial margins. Take the cycle C_{18}, add one self-loop at each node, and set $P = P_{\mathrm{g}} = (A + I)/3$ and $L_{\mathrm{g}} = I - P$. In cyclic order, use

$$\begin{aligned}
y^{\mathrm{A}}: &\quad 111011001110010000, \\
y^{\mathrm{B}}: &\quad 011001100111101000,
\end{aligned}$$

where 1 denotes $+1$ and 0 denotes -1. Each labeling has nine nodes per class and eight cross-class edges. Let k_i count the two non-self neighbors whose labels differ from y_i; both labelings have counts $(4, 12, 2)$ for $k_i = 0, 1, 2$.

For cyclic indices, define $c_y(r) = \sum_{i=0}^{17} y_i y_{i+r}$, with indices modulo 18. Direct integer calculation gives the same autocorrelation for both labelings:

$$\begin{aligned}
(c_y(r))_{r=0}^{17} = (&18, 2, -6, -2, 2, 2, -6, -6, 2, \\
&6, 2, -6, -6, 2, 2, -2, -6, 2).
\end{aligned}$$

For the unitary discrete Fourier transform $\widehat{y}_k = 18^{-1/2} \sum_j y_j e^{-2\pi \mathrm{i} kj/18}$,

$$|\widehat{y}_k|^2 = \frac{1}{18} \sum_{r=0}^{17} c_y(r) e^{2\pi \mathrm{i} kr/18}.$$

Thus the squared Fourier amplitudes agree. The Fourier vectors diagonalize the cycle Laplacian (Shuman et al., 2013); summing over repeated eigenvalues proves $p_{y^{\mathrm{A}}}(\mu) = p_{y^{\mathrm{B}}}(\mu)$ at every frequency.

For $x = y + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, the calibrated prediction uses $H(\tau)x$, where $H(\tau) = ((1-\tau)I + \tau P)P$. Each row of $H(\tau)$ has coefficients

$$\frac{1}{9}\big(\tau,\, 3-\tau,\, 3,\, 3-\tau,\, \tau\big).$$

These correspond to offsets $(-2, -1, 0, 1, 2)$. Taking $z = Px$ in Step 1 gives $m = Py$ and $\Sigma = \sigma^2 P^2$. The marginal noise standard deviation $s(\tau) = \sigma \|H(\tau)_{i:}\|_2$ is independent of i. At zero, $s(0) = s = \sigma/\sqrt{3}$ and $y_i(Py)_i = 1 - 2k_i/3$. Thus the empirical distributions of the initial standardized margins $\gamma_i(0) = (1 - 2k_i/3)/s$ agree. In particular,

$$R_{y^{\mathrm{A}}}(0) = R_{y^{\mathrm{B}}}(0) = \frac{4\Phi(-1/s) + 12\Phi(-1/(3s)) + 2\Phi(1/(3s))}{18}. \tag{21}$$

Step 3: opposite risk derivatives. The row coefficients give $(s^2)'(0) = -4\sigma^2/27$ and hence $s'(0)/s = -2/9$. Substituting $m = Py$ and $\Sigma = \sigma^2 P^2$ into Equation (20), with $L_{\mathrm{g}} = I - P$, gives

$$\begin{aligned} \gamma_i'(0) &= \frac{y_i[(P^2 - P)y]_i}{s} - \gamma_i(0)\frac{s'(0)}{s} \\ &= \frac{y_i}{s}\left[(P^2 y)_i - \frac{7}{9}(Py)_i\right] =: \frac{h_i}{s}. \end{aligned} \tag{22}$$

The exact group sums are

	$k_i = 0$	$k_i = 1$	$k_i = 2$
node count	4	12	2
$27 \sum_{i:k_i=k} h_i$ (y^{A})	-12	12	-4
$27 \sum_{i:k_i=k} h_i$ (y^{B})	0	-12	8

Writing $u = \varphi(1/s)$ and $v = \varphi(1/(3s))$, Equation (12) yields

$$R'_{y^{\mathrm{A}}}(0) = \frac{6u - 4v}{243s}, \qquad R'_{y^{\mathrm{B}}}(0) = \frac{2v}{243s} > 0. \tag{23}$$

Since $u/v = \exp(-4/(3\sigma^2))$, the first derivative is negative whenever

$$0 < \sigma < \sqrt{\frac{4}{3\log(3/2)}}.$$

Differentiability therefore implies that sufficiently small positive τ reduces risk for y^{A}, whereas sufficiently small negative τ reduces risk for y^{B}. At $\sigma = 1$, the common initial risk is approximately 0.276947, and the derivatives are approximately -0.005821 and 0.004814.

Let $w_i = \varphi(\gamma_i(0))$ and $v_i = \gamma_i'(0)$, and use bars for uniform node averages. Defining $\mathrm{Cov}_i(w, v) = N^{-1} \sum_i (w_i - \bar{w})(v_i - \bar{v})$ gives

$$R'_y(0) = -\frac{1}{N}\sum_i w_i v_i = -\bar{w}\,\bar{v} - \mathrm{Cov}_i(w, v). \tag{24}$$

The common initial margin distribution fixes $\bar{w}$. The average response also agrees, since

$$\bar{v} = \frac{y^\top (P^2 - \frac{7}{9}P)y}{Ns} = \frac{1}{Ns}\sum_\mu \left[(1-\mu)^2 - \frac{7}{9}(1-\mu)\right] p_y(\mu), \qquad s = \frac{\sigma}{\sqrt{3}}. \tag{25}$$

Thus $\bar{w}$ and $\bar{v}$ agree between the tasks, while their covariance differs and produces the opposite derivatives in Equation (23).

Step 4: arbitrarily large connected graphs. Repeat each length-18 labeling r times around C_{18r} for any positive integer r. Its cyclic autocorrelation is r times the base autocorrelation at the lag modulo 18, so the two repeated labelings again have identical spectral energy. Local label patterns, row noise variance, and group proportions are unchanged. Their average initial risks and risk derivatives therefore remain those above, completing the proof. □

Corollary 1 (Reconstruction and classification can favor different directions)**.** *For the constructed pair on C_{18r}, let $M_y(\tau) = N^{-1}\mathbb{E}\|H(\tau)x - y\|_2^2$, where $N = 18r$. If*

$$\frac{\sqrt{2}}{3} < \sigma < \sqrt{\frac{4}{3\log(3/2)}},$$

then

$$M'_{y^{\mathrm{A}}}(0) = M'_{y^{\mathrm{B}}}(0) < 0, \qquad R'_{y^{\mathrm{A}}}(0) < 0 < R'_{y^{\mathrm{B}}}(0). \tag{26}$$

Thus sufficiently small positive calibration reduces reconstruction error in both tasks while reducing classification risk only in y^{A} and increasing it in y^{B}.

Proof of Corollary 1. Write $G = P^2 - P$, so $H(\tau) = P + \tau G$. Zero-mean noise with covariance $\sigma^2 I$ gives

$$M_y(\tau) = \frac{1}{N}\|(P - I)y + \tau G y\|_2^2 + \frac{\sigma^2}{N}\|P + \tau G\|_F^2. \tag{27}$$

For either labeling in Step 2, the label patterns and propagation coefficients yield

$$\frac{\langle (P - I)y, Gy\rangle}{N} = \frac{4}{243}, \qquad \frac{\langle P, G\rangle_F}{N} = -\frac{2}{27}.$$

These normalized quantities are unchanged by periodic repetition. Differentiating Equation (27) at zero gives

$$\begin{aligned} M'_y(0) &= \frac{2}{N}\langle (P - I)y, Gy\rangle + \frac{2\sigma^2}{N}\langle P, G\rangle_F \\ &= \frac{8 - 36\sigma^2}{243} < 0 \qquad \text{when } \sigma > \frac{\sqrt{2}}{3}. \end{aligned} \tag{28}$$

Combining this with Equation (23) proves the derivative claims. Differentiability gives the strict risk changes for sufficiently small positive τ. Proposition 2 also implies a decrease in expected Dirichlet energy for $0 < \tau < 1$. □

B Additional Experimental Results

We first detail the motivation diagnostics, then compare alternative DPC and SSC designs and examine the learned calibrations. Initialization sensitivity and large-scale resource measurements complete the analysis.

B.1 Motivation Diagnostics

Degree-matched messages. Figure 1(a) examines whether neighborhood degree contrast remains associated with message changes after matching receiver-degree distributions. We apply symmetric GCN and mean propagation to the same PubMed features X, with each feature row normalized to unit ℓ_2 norm. The diagnostic uses no training or labels. With $\widetilde{\mathcal{N}}_i$ including the receiver's self-loop, define

$$u_i = \frac{1}{\widetilde{d}_i}\sum_{j\in\widetilde{\mathcal{N}}_i} |\log \widetilde{d}_j - \log \widetilde{d}_i|,$$

$$r_i = \frac{\|[(P_{\mathrm{g}} - P_{\mathrm{m}})X]_{i:}\|_2}{\|(P_{\mathrm{g}}X)_{i:}\|_2 + \|(P_{\mathrm{m}}X)_{i:}\|_2 + 10^{-12}}.$$

Here u_i is the neighborhood degree contrast, averaging absolute log-degree differences between receiver i and its neighbors; r_i measures the relative message change. Within each exact receiver-degree stratum of size n_d, we divide the percentiles $(\mathrm{rank}(u_i) - 0.5)/n_d$ into 12 equal-width bins,

using average ranks for ties. We retain strata with at least five nodes per bin and weight each node by m_d/n_{db}, where n_{db} is its degree–bin count and $m_d = \min_b n_{db}$. Thus all bins have identical weighted receiver-degree distributions. Density, medians, and 10th–90th percentiles use these weights.

Matching retains 97.55% of the nodes. The weighted median message change rises from 0.098 in the lowest contrast bin to 0.334 in the highest. This association shows that neighborhoods with matched receiver-degree distributions can still differ substantially in their sensitivity to degree-based message weighting.

State calibration. Figure 1(b) examines how SSC adjusts the degree-calibrated state on Cora, Minesweeper, and Tolokers. At each of five runs' best-validation checkpoints, we record the coefficient and relative Dirichlet-energy change in every propagation layer; points are run–layer observations and bars show their 10th–90th percentiles. Cora remains near identity, Minesweeper predominantly sharpens, and Tolokers predominantly smooths. These observations illustrate task-dependent adjustments learned by SSC; Appendix B.3 compares its design choices, and Appendix B.4 extends the behavior analysis.

B.2 DPC Design

Table 6 examines the receiver-dependent term, coordinate centering, and node–topology correspondence. *Graph-wide* sets $\lambda_l = 0$. *Uncentered* replaces t_i with $q_i / \max_j |q_j|$. *Shuffled-centered* deterministically permutes q_i across nodes before applying the same weighted centering and normalization; it retains the parameter count, bound $|t_i| \leq 1$, and weighted orthogonality while changing the topology assignment. *Full DPC* uses the original centered coordinate. All variants disable SSC and use the component-ablation settings; the graph-wide and full DPC results are the corresponding ablation entries.

Table 6: DPC coordinate controls (%, five-run mean ± standard deviation).

Coordinate	Computer	WikiCS	Minesweeper	Ratings	Roman	Questions	Tolokers
Metric	Accuracy	Accuracy	ROC-AUC	Accuracy	Accuracy	ROC-AUC	ROC-AUC
Graph-wide	94.01±0.17	80.71±0.37	97.32±0.39	55.28±0.17	91.59±0.25	79.08±0.52	86.37±0.51
Uncentered	93.95±0.06	80.47±0.35	97.51±0.45	54.92±0.47	91.62±0.10	79.04±0.52	86.36±0.52
Shuffled-centered	93.97±0.12	80.56±0.43	97.50±0.73	55.45±0.29	91.63±0.23	79.02±0.54	86.33±0.42
Full DPC	94.20±0.10	80.90±0.28	97.73±0.38	55.42±0.38	91.58±0.22	79.06±0.54	86.41±0.56

The receiver-dependent term improves the mean over graph-wide calibration by 0.19, 0.19, and 0.41 points on Computer, WikiCS, and Minesweeper, respectively. Full DPC also exceeds both the uncentered and shuffled-centered alternatives on these tasks. These comparisons favor the centered construction and its original topology assignment. Differences are smaller on the remaining tasks, where shuffled-centered calibration slightly leads on Ratings and Roman, and graph-wide calibration leads on Questions.

Centering separates the shared correction from node-dependent variation (Appendix A.2). Under the coefficient bounds used here, it changes the feasible parameterization; shuffling followed by recentering tests a construction with reassigned coordinates. Together, the comparisons support the complete DPC design on several tasks, with benefits depending on the graph.

B.3 SSC Design

Table 7 examines three choices: learning the sign, sharing the coefficient across nodes and features, and learning calibration strength instead of applying a fixed propagation step. All controls use the same DPC-GCN backbone, splits, epoch budgets, and paired backbone initializations. Learned variants retain the SSC learning-rate multipliers in Appendix C.3; results use five runs with best-validation checkpoint selection.

Let $C_l \in \mathbb{R}^{N \times d_l}$ denote the state at the SSC input. SSC applies $C_l - \tau_l L_g C_l$ with one shared scalar. *Smooth-only* and *sharp-only* retain $\tau_l = \tanh(\theta_l)$, initialize $\theta_l = 0$, and project it onto the

nonnegative or nonpositive half-line after each update. *Feature-wise* uses

$$C_l - (L_{\mathrm{g}} C_l)\,\mathrm{Diag}(\tanh \boldsymbol{\theta}_l), \qquad \boldsymbol{\theta}_l \in \mathbb{R}^{d_l},$$

with one coefficient per feature shared across nodes. *Node-wise* uses a shared feature-driven gate:

$$C_l - \mathrm{Diag}(\boldsymbol{\tau}_l(C_l)) L_{\mathrm{g}} C_l, \qquad \tau_{l,i}(C_l) = \tanh(C_{l,i:} w_l + b_l),$$

where $w_l \in \mathbb{R}^{d_l}$ and $b_l \in \mathbb{R}$. Both variants initialize calibration parameters at zero, apply no weight decay to them, and add the same single SpMM as SSC. *Fixed propagation* applies $P_{\mathrm{g}} C_l$ at every layer, equivalent to $\tau_l = 1$.

Table 7: SSC direction, granularity, and fixed-propagation controls (%). Bold marks the highest mean per dataset.

Variant	Ratings	Minesweeper	Roman	Questions	Tolokers
Metric	Accuracy	ROC-AUC	Accuracy	ROC-AUC	ROC-AUC
Smooth-only	55.17±0.22	97.82±0.35	91.59±0.17	79.03±0.54	86.58±0.52
Sharp-only	55.06±0.50	98.66±0.25	91.87±0.50	79.87±0.62	86.44±0.55
Fixed propagation	48.68±0.44	57.91±1.49	66.65±0.42	65.75±1.48	81.62±0.69
Feature-wise	55.08±0.37	98.24±0.16	91.96±0.25	**79.99±0.55**	86.48±0.51
Node-wise	54.45±0.77	98.34±0.42	91.88±0.32	78.02±0.83	85.00±0.84
SSC	**55.55±0.39**	**98.73±0.19**	**91.97±0.22**	79.89±0.63	**86.62±0.53**

Direction. Sharp-only exceeds smooth-only by 0.84 points on both Minesweeper and Questions, whereas smooth-only is 0.14 points higher on Tolokers. On Amazon-Ratings, SSC exceeds both one-sided variants. The relative performance of the constrained directions thus varies across tasks, supporting a learnable sign. Appendix A.5 provides the theoretical motivation: the beneficial local classification direction can differ even when label spectral energies and initial margin distributions agree.

Granularity. SSC has the highest mean on four datasets; feature-wise calibration is 0.10 points higher on Questions and nearly matches SSC on Roman. Node-wise calibration is lower on all five datasets. Finer coefficients therefore offer no consistent gain under these settings, supporting the compact shared scalar. Appendix A.3 establishes the node-sharing condition for a scalar spectral response; the feature-wise comparison supplies the empirical basis for also sharing across channels.

Fixed propagation. An extra full P_{g} step is lower than SSC on all five datasets, reaching 57.91 versus 98.73 on Minesweeper. This control fixes both direction and strength at $\tau = 1$, with response $g_1(\mu) = 1-\mu$ (Appendix A.4). Learned SSC instead permits near-identity, smoothing, and sharpening responses. The comparison favors adjustable calibration over a fixed additional propagation step.

B.4 Learned Calibration

Message weights. We examine how learned receiver coefficients combine with source–receiver degree relations to change individual messages. Operators are reconstructed from topology and validation-selected layer coefficients, using DPC-GCN on the eight homophilous graphs and DPC-GCN-S on the five heterophilous graphs. The analysis covers 325 run–layer states across five runs and all layers.

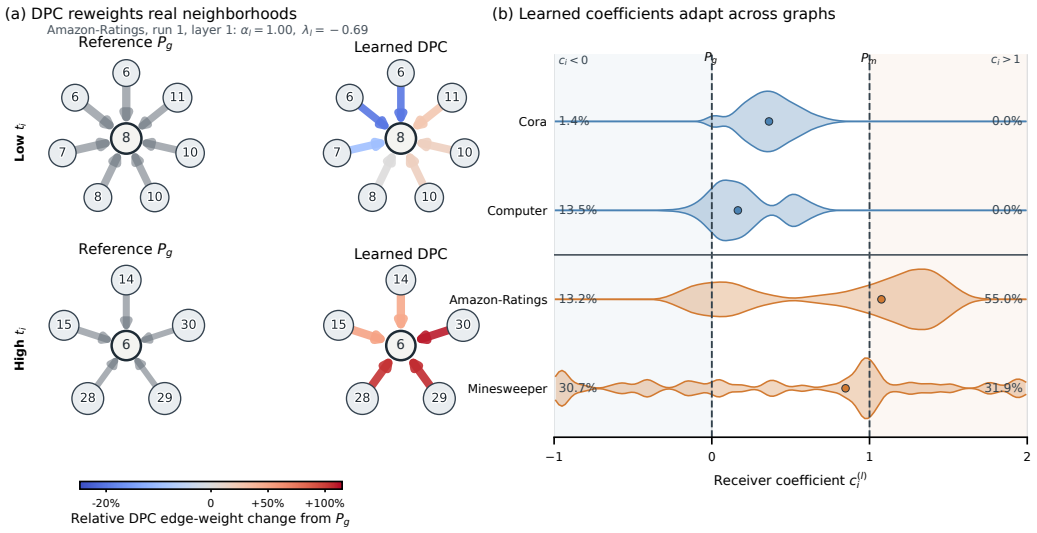

Figure 4: Learned DPC reweighting. (a) Two Amazon-Ratings neighborhoods: node labels and sizes encode effective degree, edge widths show weights, and colors indicate changes from P_{g} (blue: decreases; red: increases). (b) Receiver-coefficient distributions; percentages denote coefficients outside $[0, 1]$.

In Figure 4(a), the relative weight change $c_i \kappa_{ij}$ combines the learned receiver coefficient with the directed degree-potential contrast. The displayed low-q_i neighborhood strengthens messages from higher-degree sources and weakens those from lower-degree sources. In the high-q_i neighborhood, every displayed source has a higher degree than the receiver and its message is strengthened. Degree potential thus links the learned reweighting to concrete source–receiver relations.

Panel (b) shows interpolation and extrapolation relative to P_{g} and P_{m}, corresponding to coefficients inside and outside $[0, 1]$. Across the 13 datasets, extrapolation is most visible on Amazon-Ratings, Minesweeper, and Roman-Empire. The sign of an edge weight is determined by $1 + c_i \kappa_{ij}$. These observations show how training uses the available correction range; the design comparisons in Appendix B.2 assess its predictive value.

State responses. Table 8 records learned coefficients and energy changes from independent diagnostic runs at the main-result configurations, using validation-selected checkpoints. Minesweeper, Roman-Empire, and Questions learn negative coefficients in all recorded layers and increase Dirichlet energy; Tolokers predominantly uses positive coefficients and reduces it. Amazon-Ratings uses both signs. The observed energy responses agree with Proposition 2 and illustrate the different adjustments learned across tasks.

Table 8: Learned SSC coefficients and energy responses. Observations count run-layer coefficients; $\Delta E/E$ is the mean relative Dirichlet-energy change.

Dataset	Observations	Mean τ	Negative τ	Mean $\Delta E/E$
Amazon-Ratings	15	-0.036	66.7%	$+7.3\%$
Minesweeper	70	-0.644	100.0%	$+160.9\%$
Roman-Empire	45	-0.528	100.0%	$+124.3\%$
Questions	65	-0.272	100.0%	$+61.4\%$
Tolokers	25	$+0.220$	8.0%	-31.2%

Separately, the full models' main-result checkpoints give dataset-averaged mean $|\tau|$ values of 0.072 on homophilous graphs and 0.346 on heterophilous graphs. The stronger learned calibration in the latter group accompanies the more consistent SSC gains in the component ablations (Tables 4–5).

B.5 Initialization Sensitivity

We examine whether learned DPC coefficients remain sensitive to initialization by varying α_0 or λ_0 one at a time over the values in Figure 5, while training both coefficients throughout each run. WikiCS and PubMed use their selected DPC-GCN configurations; Amazon-Ratings and Roman-Empire use their selected DPC-GCN-S configurations. All other hyperparameters remain fixed within each sweep. Each point reports the five-run mean test score at the best-validation checkpoints relative to the corresponding reference mean.

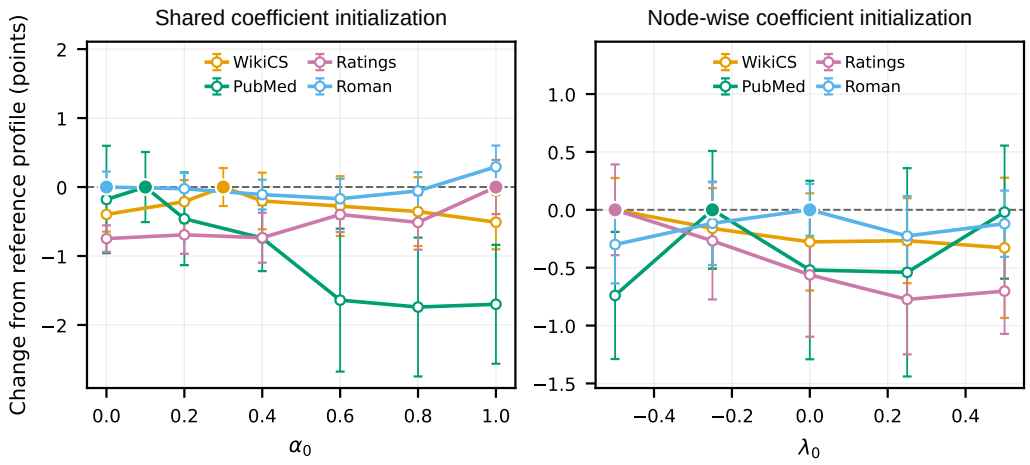

Figure 5: Test-score sensitivity to DPC initialization. Values show changes in percentage points from the reference mean; filled markers identify reference settings. Error bars show the standard deviation of each configuration over five runs.

Figure 5 shows that sensitivity to initialization varies across datasets and coefficients. WikiCS and Roman-Empire exhibit relatively small changes over the tested ranges, whereas PubMed is more sensitive to α_0, with a decrease of approximately 1.6–1.7 percentage points for $\alpha_0 \geq 0.6$. Varying λ_0 changes the mean score by less than one percentage point on each of the four datasets. These results indicate that performance remains relatively stable across a range of initializations on several tasks, while initialization selection remains useful for more sensitive cases.

B.6 Large-Scale Training Resources

Table 9 complements the large-scale predictive results with per-epoch training time and peak GPU memory at each method's main-result configuration. Appendix C.2 provides hardware details.

Table 9: Per-epoch training time and peak GPU memory for the large-scale main-result configurations.

	ogbn-arxiv		ogbn-products		Pokec	
Model	Train/Epoch (s)	Mem. (GiB)	Train/Epoch (s)	Mem. (GiB)	Train/Epoch (s)	Mem. (GiB)
Polynormer	0.859	13.27	18.556	43.90	20.360	72.68
DUALFormer	0.823	9.93	22.500	71.73	126.163	73.09
GT-SNT	0.569	16.37	4.497	27.42	3.962	38.06
Tuned-GCN	1.220	15.73	21.178	23.18	17.961	77.82
Tuned-GraphSAGE	0.427	7.16	20.397	17.58	12.116	53.45
Tuned-GAT	0.700	24.94	21.996	38.22	19.412	72.06
GAMMA	0.258	1.70	2.348	53.29	18.487	9.69
DPC-GCN	0.889	8.64	10.308	17.64	17.288	18.32
DPC-GCN-S	1.155	11.60	17.093	33.24	35.148	32.50

On ogbn-products, DPC-GCN uses 10.308 seconds per epoch and 17.64 GiB, versus 18.556 seconds and 43.90 GiB for Polynormer: $1.80\times$ faster epochs and 60% lower peak memory, alongside the 2.27-point accuracy gap in Table 3. On Pokec, DPC-GCN-S achieves the highest accuracy with 32.50 GiB versus Polynormer's 72.68 GiB, while taking longer per epoch (35.148 versus 20.360 seconds). These tasks illustrate different balances between predictive performance, memory, and epoch time.

C Experimental Details

C.1 Datasets and Evaluation

We evaluate our methods on 16 node classification benchmarks, including eight homophilous graphs, five heterophilous graphs, and three large-scale graphs. Table 10 summarizes their statistics, data splits, and evaluation metrics.

Table 10: Dataset statistics and evaluation splits.

Dataset	Nodes	Edges	Features	Classes/Tasks	Train	Valid/Test	Metric
Medium-scale graphs							
Amazon-Computer	13,752	505,474	767	10	8,251	2,750/2,751	Acc.
Amazon-Photo	7,650	245,812	745	8	4,590	1,530/1,530	Acc.
Coauthor-CS	18,333	182,121	6,805	15	10,999	3,666/3,668	Acc.
Coauthor-Physics	34,493	530,417	8,415	5	20,695	6,898/6,900	Acc.
WikiCS	11,701	442,907	300	10	580	5,274/5,847	Acc.
Cora	2,708	13,264	1,433	7	140	500/1,000	Acc.
CiteSeer	3,327	12,431	3,703	6	120	500/1,000	Acc.
PubMed	19,717	108,365	500	3	60	500/1,000	Acc.
Amazon-Ratings	24,492	210,592	300	5	12,246	6,123/6,123	Acc.
Minesweeper	10,000	88,804	7	2	5,000	2,500/2,500	ROC-AUC
Roman-Empire	22,662	88,516	300	18	11,331	5,665/5,666	Acc.
Questions	48,921	356,001	301	2	24,460	12,230/12,231	ROC-AUC
Tolokers	11,758	1,049,758	10	2	5,879	2,939/2,940	ROC-AUC
Large-scale graphs							
ogbn-arxiv	169,343	1,166,243	128	40	90,941	29,799/48,603	Acc.
ogbn-products	2,449,029	61,859,140	100	47	196,615	39,323/2,213,091	Acc.
Pokec	1,632,803	30,622,564	65	2	816,320	408,160/408,160	Acc.

Medium-scale edge counts include the undirected support and one self-loop per node; large-scale counts use the raw benchmark graphs.

Amazon-Computer, Amazon-Photo, Coauthor-CS, and Coauthor-Physics use fixed 60%/20%/20% train/validation/test splits. WikiCS and the five heterophily benchmarks use the first five official split masks. Cora, CiteSeer, and PubMed use a fixed class-balanced split with 20 training nodes per class, 500 validation nodes, and 1,000 test nodes. The OGB datasets use their official splits (Hu et al., 2020). Pokec uses the first three released LINKX splits, each following a 50%/25%/25% ratio, with unlabeled nodes excluded (Lim et al., 2021).

For locally evaluated methods, we report mean test performance and standard deviation over five runs on medium-scale graphs and three runs on large-scale graphs. Configurations are selected using validation performance, and test scores are obtained from each run's best-validation checkpoint. We use ROC-AUC for Minesweeper, Questions, and Tolokers, and accuracy for the remaining datasets.

C.2 Hardware

Medium-scale experiments are conducted on an NVIDIA GeForce RTX 5090 with 32 GB of GPU memory, and large-scale experiments on an NVIDIA A800 with 80 GB of GPU memory. Efficiency results use the configurations associated with each method's main predictive results.

C.3 Model Details

Algorithm 1 summarizes the DPC layer and its optional SSC step on a fixed graph. Graph-only quantities are cached and shared across layers; the remaining backbone operations follow the configurations below.

Algorithm 1 DPC layer with optional SSC.

Require: Graph G, layer input $H^{(l)}$, parameters $W_l, \alpha_l, \lambda_l, \theta_l$, SSC flag
Ensure: Calibrated layer state $\widehat{B}^{(l)}$

if graph quantities are not cached **then**
  Construct P_{g} and $\Delta = P_{\mathrm{m}} - P_{\mathrm{g}}$ using Eq. (1)
  Compute q and t using Eqs. (6)–(7)
  Cache sparse P_{g}, Δ and node coordinate t
end if
$Z^{(l)} \leftarrow H^{(l)} W_l, \quad c^{(l)} \leftarrow \alpha_l \mathbf{1} + \lambda_l t$
$P_{ij}^{(l)} \leftarrow (P_{\mathrm{g}})_{ij} + c_i^{(l)} \Delta_{ij}$ on the existing sparse support
$B^{(l)} \leftarrow \mathrm{SpMM}(P^{(l)}, Z^{(l)})$
Add the configured backbone bias and linear residual to $B^{(l)}$
$\widehat{B}^{(l)} \leftarrow B^{(l)}$
if SSC is enabled **then**
  $\tau_l \leftarrow \tanh(\theta_l)$
  $\widehat{B}^{(l)} \leftarrow (1 - \tau_l) B^{(l)} + \tau_l \, \mathrm{SpMM}(P_{\mathrm{g}}, B^{(l)})$
end if
return $\widehat{B}^{(l)}$ to the subsequent backbone operations

SpMM denotes sparse–dense matrix multiplication. SSC acts on the state after linear additions and before subsequent normalization and activation, using the cached P_{g}.

Table 11 lists the dataset-specific configurations of DPC-GCN and DPC-GCN-S. Both denotes their shared backbone and DPC settings on medium-scale graphs; large-scale configurations are listed separately where the variants differ. The SSC learning-rate multiplier applies only to DPC-GCN-S.

Table 11: Dataset-specific configurations for DPC-GCN and DPC-GCN-S.

Dataset	Variant	H	L	D	Epochs	LR	WD	α_0	λ_0	m_τ	Backbone
Amazon-Computer	Both	512	4	0.40	1000	10^{-3}	5×10^{-5}	0.8	0	1	LN
Amazon-Photo	Both	128	5	0.40	1000	10^{-3}	5×10^{-5}	1.0	0	0.1	LN+R
Coauthor-CS	Both	512	1	0.40	1500	10^{-3}	5×10^{-4}	0.5	0	0.1	LN+R
Coauthor-Physics	Both	64	2	0.50	1500	10^{-3}	5×10^{-4}	0.0	-0.5	1	LN+R
WikiCS	Both	128	2	0.70	1000	10^{-3}	0	0.3	-0.5	0.25	LN
Cora	Both	1024	3	0.40	500	10^{-3}	5×10^{-4}	0.5	0.5	0.1	–
CiteSeer	Both	2048	2	0.50	500	10^{-3}	10^{-2}	0.0	-0.25	0.25	–
PubMed	Both	256	2	0.80	500	5×10^{-3}	5×10^{-4}	0.1	-0.25	0.25	–
Amazon-Ratings	Both	512	3	0.65	2500	10^{-3}	0	1.0	-0.5	0.1	BN+R
Minesweeper	Both	64	14	0.20	2000	10^{-2}	0	1.0	0	0.5	BN+R
Roman-Empire	Both	256	9	0.40	2500	5×10^{-3}	0	0.0	0	1	BN+R+P
Questions	Both	512	13	0.30	1500	3×10^{-5}	0	0.0	0	64	R+P
Tolokers	Both	128	5	0.50	800	10^{-3}	0	0.0	-0.5	4	BN+R
ogbn-arxiv	DPC-GCN	512	5	0.50	2000	5×10^{-4}	5×10^{-4}	0	0	–	BN+R
ogbn-arxiv	DPC-GCN-S	512	5	0.50	2000	5×10^{-4}	5×10^{-4}	0.25	0.5	0.25	BN+R
ogbn-products	DPC-GCN	128	3	0.40	500	3×10^{-3}	0	0.25	0	–	LN
ogbn-products	DPC-GCN-S	256	5	0.50	500	3×10^{-3}	0	0.25	0	2	LN
Pokec	DPC-GCN	256	7	0.20	2000	5×10^{-4}	0	0.5	0.25	–	BN+R
Pokec	DPC-GCN-S	256	9	0.20	2000	5×10^{-4}	0	0.5	0.25	1	BN+R

Here H, L, and D denote hidden width, propagation depth, and feature-dropout rate. LN, BN, R, and P denote LayerNorm, BatchNorm, linear residual, and input projection; – means absent or not applicable.

On medium-scale benchmarks, we select $\alpha_0 \in \{0, 0.1, \ldots, 1.0\}$ and $\lambda_0 \in \{-0.5, -0.25, 0, 0.25, 0.5\}$ using mean validation performance. The SSC learning-rate multiplier is selected from $m_\tau \in \{0.1, 0.25, 0.5, 1, 2, 4\}$, with $\{4, 16, 64\}$ used on Questions.

DPC coefficients are initialized using the values in Table 11, and SSC coefficients are initialized to zero. We project α_l onto $[0, 1]$ after each update and parameterize $\lambda_l = \tanh(\rho_l)$ and $\tau_l = \tanh(\theta_l)$ with unconstrained ρ_l and θ_l. The multiplier m_τ scales the learning rate of the SSC coefficients.

We use full-graph training on medium-scale benchmarks and ogbn-arxiv. For ogbn-products, training uses ten graph partitions. Pokec uses batches of 550,000 training nodes with full-graph evaluation. In subgraph training, κ_{ij} and t_i retain their full-graph values, while P_{g} is normalized on each induced subgraph and $\Delta_{ij} = (P_{\mathrm{g}})_{ij}\kappa_{ij}$ on its edges. The endpoint identities and orthogonal decomposition refer to the full-graph operators; subgraph training reuses the resulting topology coordinates with subgraph-normalized propagation.

C.4 Baseline Configurations

We use published configurations or released implementations for the baselines and perform dataset-specific hyperparameter selection where needed, following the evaluation rules in Appendix C.1. The following paragraphs summarize the configuration sources and search ranges. Large-scale baseline configurations follow the corresponding papers or released implementations.

Tuned GNNs (Luo et al., 2024). The three Tuned-GNN baselines follow the dataset-specific configurations of the Classic GNN study and use a common PyG implementation (Fey & Lenssen, 2019). Tuned-GAT uses GAT-sep on the five heterophily benchmarks (Platonov et al., 2023).

CPF (Li et al., 2025). We use the official implementation with structure-aware and feature-aware filtering and variation-neighborhood coarsening at a reduction ratio of 0.5. Starting from released configurations, we perform validation-based coordinate search over hidden width $\{64, 128, 256\}$, polynomial order $\{5, 10, 20\}$, feature dropout $\{0, 0.3, 0.5, 0.7\}$, and filter dropout $\{0, 0.3, 0.5\}$. MLP learning rates are $\{0.001, 0.005, 0.01, 0.05\}$, with 0.1 additionally considered for filtering branches. MLP weight decay uses $\{0, 0.0005, 0.005\}$, and filter weight-decay refinement uses $\{0, 0.0005\}$. Training budgets follow Table 11.

H2GCN (Zhu et al., 2020). We use hidden width 64, two propagation rounds over strict one-hop and two-hop neighborhoods, and a 1,000-epoch budget. PubMed, Roman-Empire, and Amazon-Ratings use their matching public configurations; the remaining ten datasets select from seven candidate configurations by mean validation performance. Table 12 lists the final settings.

Table 12: Final H2GCN configurations.

Dataset	Dropout	LR	WD	Activation
Amazon-Computer	0.5	10^{-2}	0	ReLU
Amazon-Photo	0.5	10^{-2}	5×10^{-5}	identity
Coauthor-CS	0.5	10^{-2}	0	ReLU
Coauthor-Physics	0.5	10^{-2}	0	ReLU
WikiCS	0.5	10^{-2}	5×10^{-4}	ReLU
Cora	0.5	10^{-2}	5×10^{-5}	ReLU
CiteSeer	0.5	10^{-2}	5×10^{-4}	ReLU
PubMed	0.5	10^{-2}	5×10^{-5}	ReLU
Amazon-Ratings	0.0	10^{-2}	0	ReLU
Minesweeper	0.0	10^{-2}	0	ReLU
Roman-Empire	0.5	10^{-2}	5×10^{-5}	identity
Questions	0.5	10^{-2}	5×10^{-5}	identity
Tolokers	0.0	10^{-2}	0	ReLU

Co-GNN (Finkelshtein et al., 2024). We use the official implementation with the mean and sum aggregation instances on the five heterophily benchmarks. Candidate configurations follow the published dataset-specific ranges for environment/action depth and width, learning rate, and initial temperature, with learned temperature, dropout 0.2, residual connections, and LayerNorm. Validation selection yields the mean instance on Amazon-Ratings, Minesweeper, and Tolokers, and the sum instance on Roman-Empire and Questions.

AdaGNN-S (Dong et al., 2021). We use the official symmetric-normalization variant with hidden width 128 and an L_1 coefficient of 10^{-6}. Depths $\{2, 4, 8, 16\}$ are combined with the paper and release $(\mathrm{LR}, \mathrm{dropout}, \mathrm{WD})$ profiles: $(10^{-2}, 0.5, 9\times10^{-4})$ and $(10^{-3}, 0.1, 9\times10^{-6})$, respectively. Two-layer pretraining uses up to 300 epochs, followed by up to 1,000 fine-tuning epochs with validation patience 100.

GAMMA (Ghazizadeh et al., 2025). We use powers $\{0, 1, 2\}$, hidden width 32, two layers, two routing iterations, 500 epochs, and weight decay 10^{-4}. Learning rate and dropout are selected from the paper's six candidates, combining $\{0.05, 0.01, 0.002\}$ and $\{0, 0.5\}$, respectively.

GT-SNT (Zhang et al., 2026). We use the released configurations directly for Cora, CiteSeer, PubMed, Coauthor-CS, and Coauthor-Physics. On the remaining eight datasets, mean validation performance selects one of these five complete configurations.

DUALFormer (Zhuo et al., 2025). We use the official implementation and the paper's reported structures on its seven supported datasets; these structures form the candidate set for the remaining six datasets. For settings not reported per dataset, we search feature and attention dropout over $\{0.1, 0.3, 0.5\}$, classifier and attention weight decay over $\{10^{-5}, 10^{-4}, 10^{-3}\}$, attention heads over $\{2, 4\}$, and LayerNorm on or off.

SGFormer (Wu et al., 2023b). We use the official configurations directly for Cora, CiteSeer, and PubMed. For the remaining ten datasets, validation performance selects an initial configuration from these three profiles, followed by coordinate search over learning rate, hidden width, branch-specific dropout and weight decay, and graph-branch mixing weight within the paper's ranges. GNN and attention depth, heads, and training budget remain fixed to the selected profile.

Polynormer (Deng et al., 2024). We use paper or released configurations where available. For each of Cora, CiteSeer, and PubMed, we evaluate five configurations: one reference and four variants within the ranges in Table 13. Dropout and global dropout vary together; local/global training budgets are 100/500 epochs, and the number of heads is fixed to 8.

Table 13: Parameter values used in the five Polynormer candidates for each Planetoid dataset.

Dataset	Hidden	Local layers	Global layers	Dropout	Input dropout	LR	WD
Cora	512	$\{2, 3\}$	$\{1, 2\}$	$\{0.6, 0.7\}$	$\{0.1, 0.2\}$	10^{-3}	5×10^{-4}
CiteSeer	512	$\{2, 3\}$	$\{1, 2\}$	$\{0.5, 0.6\}$	$\{0.1, 0.2\}$	10^{-3}	10^{-2}
PubMed	$\{256, 512\}$	2	$\{1, 2\}$	$\{0.6, 0.7\}$	$\{0.1, 0.2\}$	5×10^{-3}	5×10^{-4}