A Unified Uncertainty Representation for Graph Neural Networks via Doubly-Spectral Stochastic Expansion

Fred Xu[*]
Department of Computer Science
University of California, Los Angeles
fredxu@cs.ucla.edu

Thomas Markovich
Block, Inc.
tmarkovich@squareup.com

Florence Regol
Mila, Quebec AI Institute
florence.regol@mila.quebec

Yizhou Sun
Department of Computer Science
University of California, Los Angeles
yzsun@cs.ucla.edu

Abstract

Reliable deployment of graph neural networks requires calibration, out-of-distribution (OOD) detection, and robustness to distribution shift, yet existing methods address these needs with separate models and objectives. We model uncertain node embeddings as random graph signals: graph Fourier filters capture structural variation, and a scalar orthogonal-polynomial chaos coordinate captures latent stochastic variation. The resulting *doubly-spectral stochastic* (DSS) expansion supplies task-matched readouts from one representation: the mean coefficient encodes class evidence for the energy-based OOD score, the higher-order coefficients encode structured logit variation, and quadrature averaging over the chaos coordinate defines the single predictive distribution used for prediction and calibration. A capacity theorem shows that, under a full-rank feature assumption, a restricted subfamily matches the chaos coefficients of any Gaussian-latent random graph signal, with exponentially decaying truncation error under a growth condition; the task-level claims are established empirically. *DSS-GNN* has two deployment modes: standalone, or as a residual branch beside a deterministic encoder (*DSS-Hybrid*). Standalone DSS-GNN achieves the lowest Brier score among the compared uncertainty-aware baselines on all 14 node classification benchmarks without post-hoc correction; DSS-Hybrid achieves the best AUROC on most node-OOD settings, competitive cross-graph OOD detection, and the strongest shifted accuracy on all 7 GOOD concept-shift benchmarks under standard empirical risk minimization (ERM). Cross-evaluating both modes on all three tasks shows that each remains effective on the other's tasks, with documented exceptions, and yields explicit deployment guidance.

1 Introduction

Graph neural networks (GNNs) perform well on graph-structured data, yet reliable deployment in drug discovery [31], fraud detection [7], or traffic forecasting [18] requires calibrated probabilities, out-of-distribution (OOD) detection, and robustness to distribution shift. On graphs the three needs are coupled, because uncertainty depends on neighborhood structure as well as node features, and they are

[*]Work done during an internship at Block, Inc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

instances of one question: *when, and how confidently, can the model predict?* Calibration measures the agreement between confidence and accuracy in distribution (ID), OOD detection requires low confidence outside the training support, and shift robustness requires that accuracy decrease slowly as the input distribution changes. A representation that explicitly encodes predictive uncertainty addresses all three, with calibrated probabilities, OOD evidence, and a regularized residual under shift obtained from one signal. Most existing methods target one requirement: sampling-based uncertainty methods add computational overhead [10, 17, 19]; evidential models such as Graph Posterior Network (GPN) [30] provide OOD evidence but not shift-robust classification, and intrinsic uncertainty models such as G-ΔUQ [33] target calibration under shift; energy-based detectors [9, 36] provide separation scores rather than calibrated probabilities; and robust-training methods [1, 27, 38] do not model uncertainty explicitly.

Spectral GNNs learn filters along the graph-frequency axis [6]; we add a *second* spectral axis for stochastic variation. *DSS-GNN* represents each node embedding as a polynomial chaos expansion in a scalar Gaussian latent [37] and propagates the joint graph-frequency/chaos-order coefficients through a learnable doubly-spectral layer (Figure 4, Appendix H). One deterministic forward pass gives task-matched readouts: the quadrature-averaged predictive distribution, whose argmax is the point predictor, is used for prediction and calibration, with the higher-order coefficients encoding structured logit variation; the zeroth-order mean logit gives an energy score for OOD detection that retains the common-logit shifts that softmax cancels; and, paired with a deterministic encoder (*DSS-Hybrid*, e.g., GCN [15]), the DSS branch acts as a regularized low-/high-pass residual under distribution shift. DSS-GNN is thus one uncertainty representation with two deployment modes; Section 4.5 evaluates both on all three tasks and states which to prefer where. Our contributions are:

1. A *doubly-spectral stochastic representation* for GNNs with a capacity theorem: under a full-rank feature assumption, a restricted DSS subfamily matches the chaos coefficients of any Gaussian-latent random graph signal, with exponentially decaying truncation error for targets with an entire extension of Gaussian growth (Theorem 2). The theorem (Section 3.3) establishes representational capacity, consistent with the empirical plateau at P=1–2, but does not by itself guarantee calibration, OOD detection, or robustness, which are supported empirically.
2. A learnable doubly-spectral layer whose quadrature-averaged loss differs from the mean-logit loss by a curvature-weighted correction over the chaos covariance (Proposition 2), relating the chaos coefficients to calibration, bounding the gap between the quadrature predictive and the mean-logit softmax (Corollary 1), and identifying the mean-logit energy as the readout that retains common-logit shifts.
3. Empirically, the lowest Brier score among compared uncertainty-aware baselines on all 14 calibration benchmarks (standalone), the best AUROC on most node-OOD settings with competitive cross-graph OOD, and the strongest shifted accuracy on all 7 concept-shift benchmarks under ERM (hybrid), plus a cross-evaluation of both modes on all three tasks, component-wise ablations (Table 21), and explicit deployment guidance.

2 Background

Spectral GNNs and uncertainty on graphs. A graph signal on $\mathcal{G} = (V, E)$ with N nodes is a vector $x \in \mathbb{R}^N$ in the Hilbert space $\ell^2(V)$; the Laplacian L_G is self-adjoint and its eigenvectors form the graph Fourier basis [24, 29]. Spectral GNNs define graph convolutions as polynomial filters of L_G; ChebNet [6] uses $g(L_G) = \sum_{k=0}^{K} c_k T_k(\widetilde{L}_G)$ with learned coefficients c_k, Chebyshev polynomials T_k, and the rescaled Laplacian $\widetilde{L}_G = 2L_G/\lambda_{\max} - I$ ($\lambda_{\max}$ the largest eigenvalue), which keeps filtering localized and allows low-pass or high-pass responses under homophily or heterophily [2]. DSS-GNN adds a second spectral axis, chaos order, orthogonal to graph frequency, and embeds uncertainty directly into spectral filtering.

Polynomial chaos and the doubly-spectral expansion. To organize *stochastic* variation spectrally, as the graph Fourier basis organizes structural variation, we use polynomial chaos expansion (PCE) [4, 35, 37], classical in uncertainty quantification [23] but, to our knowledge, not previously combined with graph spectral filtering inside a GNN (Appendix A.1 gives a self-contained introduction). PCE represents a square-integrable random variable $Y(\omega)$ as a series in the orthonormal Hermite polynomials $\{\Psi_n(\omega)\}_{n\geq 0}$ of a Gaussian latent $\omega \sim \mathcal{N}(0, 1)$; truncating at order P gives $Y(\omega) \approx$

$\sum_{n=0}^{P} Y_n \Psi_n(\omega)$, where Y_0 is the mean and $\sum_{n=1}^{P} Y_n^2$ (the *chaos energy*) the variance. Combined with the Laplacian eigenbasis $\{u_j\}$ on $\ell^2(V)$, the Hermite basis on $L^2(\mathbb{R}, \gamma)$ (γ the standard Gaussian measure) gives a tensor-product representation: a random graph signal determined by ω is in $\ell^2(V) \otimes L^2(\mathbb{R}, \gamma)$ and admits the *doubly-spectral stochastic expansion*

$$Y(\omega) = \sum_{n=0}^{\infty} \sum_{j=1}^{N} u_j \, \hat{Y}_{n,j}^{\top} \, \Psi_n(\omega), \tag{1}$$

where $\hat{Y}_{n,j} \in \mathbb{R}^d$ is the joint coefficient at graph frequency λ_j (the j-th Laplacian eigenvalue) and chaos order n, and d is the signal dimension; $\{u_j \otimes \Psi_n\}$ is a complete orthonormal basis of this space (Remark 2), ω is scalar, and each chaos coefficient is a d-dimensional graph signal. DSS-GNN learns a truncation and a propagation rule on this tensor-product space.

3 Method

DSS-GNN instantiates the doubly-spectral expansion (1) as a trainable network by learning a coupled propagation rule for the joint coefficients $\hat{Y}_{n,j}$ through parameterized graph filters and non-intrusive chaos projection (Figure 4, Appendix H).

Intuition. A random graph signal assigns to each node a random variable with finite second moment. We model all randomness as driven by a single shared standard Gaussian factor ω: every node embedding is a deterministic function of ω, expanded in the Hermite polynomials $\Psi_n(\omega)$, which play along the stochastic axis exactly the role that the graph Fourier modes play along the structural axis. A DSS layer therefore stores, for every node, $P + 1$ coefficient vectors (one per chaos order) instead of a single hidden vector; a deterministic embedding is the special case in which only the order-0 coefficient is nonzero. Each layer decomposes the embeddings onto this basis, applies learned filters jointly per graph frequency and per chaos order (dual low/high-pass Chebyshev filters along the graph axis, per-order gates along the chaos axis), and reassembles. The core Gaussian is never materialized: because pointwise nonlinearities mix chaos orders, the layer evaluates the embedding field at S deterministic Gauss–Hermite nodes, applies the activation there, and projects back onto the Hermite basis. With $S \geq P + 1$ the polynomial filter and gate terms are integrated exactly at the gate orders used in practice ($P_g \leq 1$, Appendix A), and the nonlinearity is handled by a controlled non-intrusive projection (Remark 3), so the network is deterministic end to end: no Monte-Carlo sampling and no sampling variance. The expansion itself requires only the shared Gaussian factor and finite second moments; the exponential truncation rate of Theorem 2 additionally requires targets with an entire (analytic) Gaussian extension. Readouts then select task-appropriate functionals of the same expansion: the quadrature-averaged predictive (8) for prediction and calibration, and the mean-logit energy (9) for OOD scores.

3.1 DSS layer

Let $\mathcal{G} = (V, E)$ have $N = |V|$ nodes, features $X \in \mathbb{R}^{N \times d_{\text{in}}}$, and Laplacian L_G. DSS-GNN stacks L layers on a truncated expansion at chaos order P; the input is lifted by $H_n^{(0)} = X W_{\text{in}}^{(n)}$ with $W_{\text{in}}^{(n)} \in \mathbb{R}^{d_{\text{in}} \times d_0}$, and at layer $\ell \in \{0, \ldots, L-1\}$ the embeddings are

$$H^{(\ell)}(\omega) = \sum_{n=0}^{P} H_n^{(\ell)} \Psi_n(\omega), \qquad H_n^{(\ell)} \in \mathbb{R}^{N \times d_\ell}, \tag{2}$$

with d_ℓ the hidden width, so each node maintains $P+1$ coefficient-valued hidden states. The layer updates them along *both spectral axes*. Along the graph-frequency axis, dual Chebyshev filters propagate each coefficient through localized low-pass and high-pass branches,

$$G_{\text{lp}}(L_G) = \sum_{k=0}^{K_{\text{lp}}} c_k^{(\text{lp})} T_k(\widetilde{L}_G), \qquad G_{\text{hp}}(L_G) = \sum_{k=0}^{K_{\text{hp}}} c_k^{(\text{hp})} T_k(\widetilde{L}_G), \tag{3}$$

with filter degrees $K_{\text{lp}}, K_{\text{hp}}$ and learned coefficients $c_k^{(\text{lp})}, c_k^{(\text{hp})} \in \mathbb{R}$. Along the chaos-order axis, per-order *chaos gates* $\alpha_n(\omega), \beta_n(\omega)$ couple the two branches; they are deterministic scalars or low-order

chaos expansions of gate order P_g with learnable $a_{n,r}, b_{n,r} \in \mathbb{R}$:

$$\alpha_n(\omega) = \sum_{r=0}^{P_g} a_{n,r}\Psi_r(\omega), \qquad \beta_n(\omega) = \sum_{r=0}^{P_g} b_{n,r}\Psi_r(\omega). \tag{4}$$

Because pointwise nonlinearities mix chaos orders, the update has no closed form. A non-intrusive projection evaluates the field at S Gauss–Hermite nodes $\{(\omega_s, \mu_s)\}_{s=1}^{S}$ (nodes $\omega_s \in \mathbb{R}$, weights $\mu_s > 0$), applies the activation σ, and projects back:

$$\widetilde{H}^{(\ell+1)}(\omega_s) = \left(\sum_{n=0}^{P} \Psi_n(\omega_s) \Big(\alpha_n(\omega_s) G_{\mathrm{lp}}(L_G) H_n^{(\ell)} + \beta_n(\omega_s) G_{\mathrm{hp}}(L_G) H_n^{(\ell)} \Big) \right) W_\ell. \tag{5}$$

$$H_m^{(\ell+1)} = \sum_{s=1}^{S} \mu_s \, \sigma\big(\widetilde{H}^{(\ell+1)}(\omega_s)\big) \Psi_m(\omega_s), \qquad m = 0, \ldots, P, \tag{6}$$

with $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ trainable (quadrature details in Appendix A.1; notation in Table 30, Appendix H).

3.2 Uncertainty readout

After L layers a linear readout $W_{\mathrm{out}} \in \mathbb{R}^{d_L \times C}$ (C classes, labels $y_i \in \{1, \ldots, C\}$) produces per-order logit coefficients $Z_{i,n} := (H_n^{(L)} W_{\mathrm{out}})_i \in \mathbb{R}^C$ for node i. The mean logit $Z_{i,0}$ encodes the zeroth-order class evidence used by the energy score; the logit covariance and chaos energy are

$$\Sigma_{z_i} = \sum_{n=1}^{P} Z_{i,n} Z_{i,n}^\top \in \mathbb{R}^{C \times C}, \qquad \mathcal{E}_i = \mathrm{tr}(\Sigma_{z_i}) = \sum_{n=1}^{P} \|Z_{i,n}\|^2. \tag{7}$$

The quadrature logits $z_i^{(s)} = \sum_{n=0}^{P} Z_{i,n}\Psi_n(\omega_s)$ are deterministic evaluations of the learned expansion (no Gaussian noise is injected in training or inference) and define the readouts. The *quadrature-averaged predictive*

$$\bar{p}_i = \sum_{s=1}^{S} \mu_s \, \mathrm{softmax}(z_i^{(s)}) \in \mathbb{R}^C \tag{8}$$

is the model's predictive distribution and the point prediction is its argmax, $\hat{y}_i = \arg\max_c \bar{p}_{i,c}$, so prediction and calibration are two functionals of one distribution. Its deviation from the mean-logit softmax is bounded by the chaos covariance, $\|\bar{p}_i - \mathrm{softmax}(Z_{i,0})\|_\infty \le \frac{B}{2} \mathrm{tr}(\Sigma_{z_i})$ for a universal constant B (Corollary 1), which training keeps small (Section 3.5); Section 4.2 measures the agreement. For OOD detection, the *energy score* [20]

$$s_{\mathrm{energy}}(i) = -\log \sum_{c=1}^{C} \exp(Z_{i,0,c}), \tag{9}$$

with $Z_{i,0,c}$ the c-th mean logit, captures confidence in the mean prediction; predictive entropy and mutual information of the quadrature softmaxes are auxiliary summaries of the same readout. The OOD experiments use the energy score with graph propagation, following GNNSafe [36].

3.3 Theoretical properties

Four results connect the layer to the expansion (1); proofs are in Appendix A.

Proposition 1 (Generalization)**.** *With $P = 0$ and $P_g = 0$,* (2) *has a single deterministic coefficient $H_0^{(\ell)}$ with scalar gates, and the DSS layer reduces to a standard polynomial spectral GNN layer, recovering ChebNet, GCN, and related models as special cases.*

Theorem 1 (Doubly-spectral operator)**.** *With the pointwise nonlinearity omitted, gates of chaos degree at most P_g, states of degree at most P, and a Gauss–Hermite rule exact for all triple products $\Psi_r\Psi_n\Psi_m$ ($r \le P_g$; $n, m \le P$; $S \ge \lceil (P_g + 2P + 1)/2 \rceil$ suffices), the linearized DSS layer is a linear operator on the joint graph-frequency/chaos-order domain of* (1)*, acting frequency by frequency: the Chebyshev filters set the graph-frequency response and the chaos gates the chaos-order mixing (Appendix A.4; the omitted nonlinearity's projection error is decomposed in Proposition 3).*

Theorem 2 (Representational capacity and truncation under full-rank features)**.** *Let $\mathcal{G}$ have N nodes and features $X \in \mathbb{R}^{N\times d_{\text{in}}}$ with $\operatorname{rank}(X) = N$ (which requires $d_{\text{in}} \geq N$). With ReLU activation, a restricted subfamily of DSS-GNNs (one layer, lifted width $2d(P{+}1)$, quadrature size $S = P{+}1$, constant gates, identity low-pass filter) has the following properties. For any target $Y \in L^2(\mathbb{R}, \gamma; \mathbb{R}^{N\times d})$ and any chaos order P, some output $\widetilde{Y}_P$ matches the chaos coefficients of Y through order P; letting P and width vary, the resulting outputs are dense in $L^2(\mathbb{R}, \gamma; \mathbb{R}^{N\times d})$. If Y in addition admits an entire Gaussian extension with controlled growth (Appendix A.6), the truncation error decays exponentially:*

$$\|Y - \widetilde{Y}_P\|_{L^2(\mathbb{R},\gamma;\,\mathbb{R}^{N\times d})} \leq C_{\text{tr}}\, \beta^P \tag{10}$$

for some $C_{\text{tr}} > 0$ and $\beta \in (0, 1)$ depending on Y.

Theorems 1 and 2 are capacity statements: they establish the operator structure and the representational capacity of the layer and identify no inherent expressivity obstruction, but they do not by themselves guarantee calibration, OOD detection, or robustness, which are empirical claims supported task by task in Section 4. The full-rank case requires $d_{\text{in}} \geq N$; for $\operatorname{rank}(X) < N$, the common case (Cora has 1,433 features for 2,708 nodes), Remark 5 shows that the same construction still matches every truncated target whose chaos-coefficient columns lie in $\operatorname{range}(X)$, under one positivity condition satisfiable by appending a constant feature column.

Proposition 2 (Stochastic and mean-logit readouts)**.** *Assume the number of Gauss–Hermite quadrature nodes is at least one more than the chaos order, and that the per-node loss $\ell(\cdot, y_i)$ satisfies the regularity conditions in Lemma 3. Then:*

1. *(Calibration channel.) The per-node quadrature-averaged loss $\bar{\ell}_i := \sum_{s=1}^{S} \mu_s\, \ell(z_i^{(s)}, y_i)$ admits the Taylor expansion $\bar{\ell}_i = \ell(Z_{i,0}, y_i) + \frac{1}{2}\operatorname{tr}(\mathcal{H}_i\, \Sigma_{z_i}) + O\big(\max_s \|z_i^{(s)} - Z_{i,0}\|^3\big)$, where $\mathcal{H}_i = \nabla_z^2 \ell(Z_{i,0}, y_i)$ is the loss Hessian and Σ_{z_i} is the chaos covariance* (7)*.*
2. *(OOD channel.) Shifting every class logit by the same constant leaves the softmax output, and hence any softmax-derived summary such as predictive entropy or mutual information, unchanged; by contrast, the energy score* (9) *decreases by exactly that constant.*

Calibration: the curvature-weighted term $\frac{1}{2}\operatorname{tr}(\mathcal{H}_i \Sigma_{z_i})$ is the leading-order difference between the quadrature-averaged and the mean-logit objectives; it weights the chaos covariance by the loss curvature and, for cross-entropy, is at most $\frac{1}{4}\mathcal{E}_i$, the quantity that (11) regularizes; no post-hoc temperature scaling is used. The same expansion applied to the softmax readout bounds the gap between $\bar{p}_i$ and $\operatorname{softmax}(Z_{i,0})$ by the chaos covariance (Corollary 1), which the objective keeps small; the calibration gains of $P > 0$ therefore arise through training rather than a readout-time correction, and the two trained readouts agree closely (Section 4.2). *OOD detection:* the energy score is linear in a common logit shift, while softmax-derived summaries cancel it, so an OOD input that uniformly lowers the class-evidence logits changes the energy but not the entropy; Section 4.3 confirms that detection is insensitive to the chaos order, as this readout predicts.

3.4 Hybrid deployment

Distribution shift weakens the common-evidence signal and changes which graph frequencies are predictive; like OOD detection it requires a strong deterministic predictor on seen structure paired with a regularized residual that accounts for cross-distribution variability. *DSS-Hybrid* adds DSS-GNN as a residual branch beside a deterministic base encoder with one shared class readout, $z_{i,0}^{\text{hyb}} = z_i^{\text{base}} + \gamma_{\text{res}} Z_{i,0}$, with $z_i^{\text{base}} \in \mathbb{R}^C$ the base logit and $\gamma_{\text{res}} \in \mathbb{R}$ a learned scale (lowercase for post-combination logits, uppercase $Z_{i,n}$ for the DSS coefficients); the quadrature logits combine the same way, $z_i^{\text{hyb},(s)} = z_i^{\text{base}} + \gamma_{\text{res}} \sum_{n=0}^{P} Z_{i,n} \Psi_n(\omega_s)$, so the readouts of Section 3.2 transfer directly. The chaos-energy regularizer keeps the residual from fitting in-distribution noise, so its dual filters retain capacity for shifts in graph-frequency response and its mean logit remains a common-evidence channel for the energy score. The base encoder, a 2-layer GCN for fair comparison with baselines, is warmed up before the DSS branch is activated. This design is an inductive bias rather than a distribution-free guarantee.

3.5 Training and downstream tasks

Both modes are trained with the chaos-energy-regularized mean-logit loss

$$\mathcal{L}_{\text{base}} = \frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{i \in \mathcal{V}_{\text{tr}}} \ell(Z_{i,0},\, y_i) \;+\; \lambda_{\text{reg}} \frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{i \in \mathcal{V}_{\text{tr}}} \mathcal{E}_i, \tag{11}$$

with $\mathcal{V}_{\text{tr}}$ the training nodes, ℓ the per-node cross-entropy, and $\lambda_{\text{reg}} \geq 0$; for DSS-Hybrid, $Z_{i,0}$ is replaced by $z_{i,0}^{\text{hyb}}$ and the regularizer applies to the DSS coefficients only. Only the mean logit is supervised, but the projection update (6) couples $Z_{i,0}$ to all chaos orders, so the higher-order coefficients are trained implicitly while the regularizer constrains their variation. By Proposition 2, $\mathcal{L}_{\text{base}}$ agrees to first order with the quadrature-averaged loss $\mathcal{L}_{\text{quad}}$ (35) of Appendix D; we use $\mathcal{L}_{\text{base}}$ throughout. The trace that Corollary 1 bounds is the regularized chaos energy $\mathcal{E}_i$; where the selected λ_{reg} is small or zero, the measured readout gap remains small (Table 27).

4 Experiments

4.1 Setup

For *calibration*, standalone DSS-GNN is compared with TFE-GNN [8] and G-ΔUQ [33] on 14 node classification benchmarks [15, 25, 26, 28], reporting accuracy and the Brier score [3] over 10 splits without post-hoc calibration. For *OOD detection*, DSS-Hybrid is evaluated on the GNNSafe benchmark [36] (nine node-OOD settings on Cora, Amazon-Photo, and Coauthor-CS, plus cross-graph Twitch and Arxiv; 3 seeds) against GNNSafe, GNNSafe++, and GPN [30] at the values reported by Wu et al. [36], and against Graph-EBM [9] (scored with its own diffusion), MC-dropout (M=20) [10] and Deep Ensembles (M=5) [17] on the same GCN backbone with the same score propagation. For *distribution shift*, DSS-Hybrid is trained with empirical risk minimization (ERM) on 7 concept-shift settings of GOOD [12] and compared with the 12 GOOD/TAR baselines [1, 27, 38] and G-ΔUQ. Point predictions are $\arg\max \bar{p}_i$ (Section 3.2); Table 1 reports the mean-logit readout, which Section 4.2 shows to agree with it. All experiments use $\mathcal{L}_{\text{base}}$ (11); in the OOD experiments DSS-Hybrid, like GNNSafe++, adds the GNNSafe++ energy-margin regularizer at per-dataset margins. Baselines, datasets, and configurations are in Appendices B–D; code is available at `https://github.com/heraclixus/DSSGNN`.

4.2 Node classification and calibration

Table 1 shows that DSS-GNN achieves the lowest Brier score among the compared uncertainty-aware baselines on all 14 datasets, homophilous and heterophilous, without post-hoc calibration, and the best accuracy on 13. The Brier gains over the deterministic $P = 0$ arm are largest on Citeseer, Texas, and Minesweeper (Table 14), consistent with the curvature-weighted correction of Proposition 2. This is not an accuracy effect: standard GNNs match or exceed DSS-GNN in accuracy on Cora and Citeseer while remaining less calibrated (Appendix F), and at $P = 0$ DSS-GNN is a competitive dual-filter spectral GNN (Proposition 1), so the gains are attributable to the chaos expansion (Table 19). Figure 1 visualizes the correction $c_i = \frac{1}{2}\,\text{tr}(\mathcal{H}_i \Sigma_{z_i})$ on Minesweeper: it is largest where stochastic variation aligns with high-curvature directions and concentrates on the hardest nodes.

Prediction and calibration use one predictive distribution. Under the full protocol with both readouts evaluated at the same checkpoints, $\arg\max \bar{p}_i$ and $\arg\max Z_{i,0}$ disagree on at most 0.21% of test nodes (exactly zero on 10 of 14 datasets), with accuracy within 0.04 points and Brier within 0.003 (Table 27, Appendix G); in hybrid mode the disagreement is 0.0000 on all 14 datasets. The agreement is structural: for $S \geq P{+}1$ Corollary 1 bounds the readout gap by $\text{tr}(\Sigma_{z_i})$, which the objective keeps small (Section 3.5), so the reported calibration is that of the distribution the predictions are drawn from.

Chaos order. Sweeping $P \in \{0, \ldots, 6, 8\}$ (Appendix E, Figure 3) improves Brier on 12 of 14 datasets at $P > 0$, and at least one of $P = 1$ or $P = 2$ is within 0.02 Brier of the best swept order on every dataset, consistent with the exponential truncation bound of Theorem 2. P is selected per dataset on validation Brier (Table 12); the larger selected orders are near-ties rather than requirements, and we recommend $P = 1$–2 as the default. Within the shared-factor family $P = 1$ is exactly a

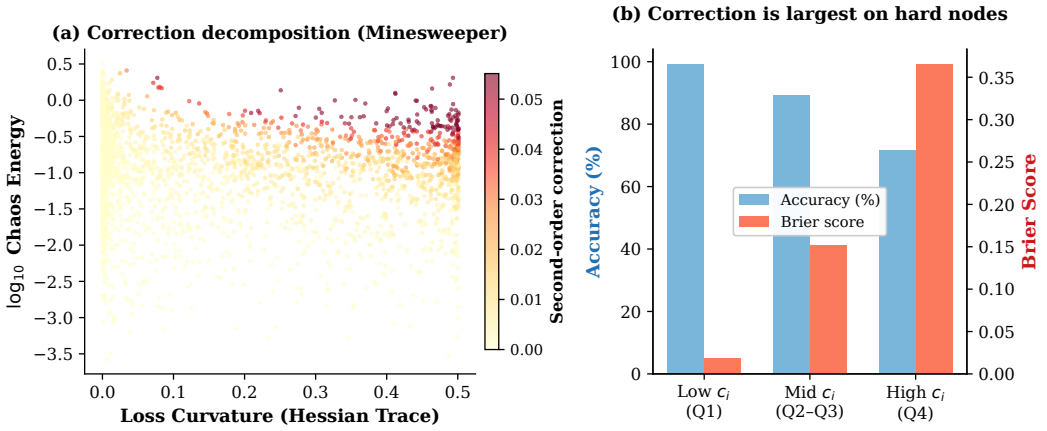

Figure 1: **(a)** Second-order correction c_i (Proposition 2) on Minesweeper. Each node is positioned by loss curvature ($\mathrm{tr}(\mathcal{H}_i)$) and chaos energy ($\mathcal{E}_i$); color shows correction magnitude. **(b)** Nodes binned by correction quartile: the correction concentrates on hard nodes (low accuracy, high Brier).

Table 1: Node classification: test accuracy (%, ↑) and Brier score (↓). Mean ± standard deviation over 10 runs. Best result per dataset in **bold**. DSS-GNN uses the per-dataset configuration of Table 12 and the mean-logit readout.

	Accuracy (%)			Brier Score		
Dataset	DSS-GNN	TFE-GNN	G-ΔUQ	DSS-GNN	TFE-GNN	G-ΔUQ
Cora	**86.63 ± 1.26**	71.25 ± 1.90	84.35 ± 1.97	**0.207 ± 0.019**	0.449 ± 0.024	0.226 ± 0.023
Citeseer	**80.20 ± 1.28**	68.87 ± 1.87	70.85 ± 2.22	**0.308 ± 0.010**	0.463 ± 0.020	0.433 ± 0.026
PubMed	**89.72 ± 0.31**	82.84 ± 0.76	88.18 ± 0.66	**0.156 ± 0.005**	0.265 ± 0.011	0.179 ± 0.008
Texas	**91.31 ± 3.44**	84.92 ± 4.26	9.02 ± 1.83	**0.240 ± 0.111**	0.255 ± 0.083	0.779 ± 0.029
Cornell	**85.11 ± 5.12**	81.28 ± 5.45	21.06 ± 3.49	**0.231 ± 0.079**	0.290 ± 0.069	0.793 ± 0.023
Wisconsin	**93.25 ± 3.39**	76.50 ± 23.95	52.25 ± 17.80	**0.104 ± 0.047**	0.292 ± 0.186	0.571 ± 0.109
Chameleon	**75.36 ± 1.19**	73.41 ± 1.53	37.31 ± 2.36	**0.339 ± 0.019**	0.365 ± 0.016	0.765 ± 0.011
Squirrel	68.06 ± 2.21	**71.56 ± 1.42**	25.88 ± 1.91	**0.394 ± 0.027**	0.395 ± 0.017	0.795 ± 0.006
CS	**96.17 ± 0.23**	95.05 ± 0.24	94.37 ± 0.23	**0.062 ± 0.004**	0.082 ± 0.005	0.085 ± 0.004
Roman-Emp.	**78.84 ± 0.61**	70.35 ± 0.60	51.24 ± 1.07	**0.300 ± 0.007**	0.414 ± 0.006	0.641 ± 0.008
Amz-Rat.	**49.72 ± 0.64**	48.79 ± 0.43	44.43 ± 0.42	**0.634 ± 0.005**	0.637 ± 0.004	0.674 ± 0.003
Minesweeper	**87.66 ± 1.10**	82.42 ± 0.33	80.22 ± 0.22	**0.169 ± 0.014**	0.248 ± 0.003	0.288 ± 0.003
Tolokers	**79.88 ± 0.49**	78.18 ± 0.10	79.75 ± 0.55	**0.268 ± 0.005**	0.382 ± 0.004	0.289 ± 0.007
Questions	**97.17 ± 0.04**	97.07 ± 0.09	97.06 ± 0.05	**0.053 ± 0.001**	0.061 ± 0.001	0.054 ± 0.001

Gaussian embedding; the higher orders add non-Gaussian structure (Citeseer Brier 0.346 to 0.308 at $P = 2$; Tables 13–14).

4.3 OOD detection

Tables 2–3 compare DSS-Hybrid with GNNSafe++ [36], the strongest OOD baseline: it has the best AUROC on 6 of 9 node-OOD cells (Cora 94.32/97.60/94.11 versus 90.62/95.56/92.75 for structure/feature/label; Amazon-Photo feature/label 99.66/97.52; Coauthor-CS label 98.07) and is within 0.13 points of the best method on the three remaining cells, where both it and GNNSafe++ exceed 99.6. On Twitch it improves over GNNSafe++ on every metric (AUROC 95.75 vs 95.36, FPR95 22.40 vs 33.57); on Arxiv it gives the best FPR95 and ID accuracy while remaining within 1.41 AUROC points of GNNSafe++ (73.36 vs 74.77). DSS-Hybrid follows the GNNSafe++ protocol (same GCN backbone configuration, energy-margin objective with its per-dataset margins, and score propagation; Appendix D), so the two differ in the DSS residual branch. Against MC-dropout and Deep Ensembles, which receive the same propagation, DSS-Hybrid has the higher AUROC on all 11 settings in one forward pass versus their 20 passes and 5 models; the published GPN values are below GNNSafe++ throughout.

Table 2: OOD detection (AUROC, %) on the nine node-OOD settings and the two cross-graph settings (AUPR, FPR95, and ID accuracy for the latter are in Table 3). GNNSafe/GNNSafe++ and GPN [30] values as reported by Wu et al. [36] on this benchmark (their Tables 1–2; their appendix lists $81.77/93.24$ for GPN on Coauthor-CS feature/label; GPN out-of-memory on Arxiv); Graph-EBM [9], MC-dropout ($M{=}20$), and Deep Ensembles ($M{=}5$) run in our pipeline on the same GCN backbone (3 seeds; the latter two with the same score propagation). Bold: best.

Model	Cora			Amazon-Photo			Coauthor-CS			Cross-graph	
	Structure	Feature	Label	Structure	Feature	Label	Structure	Feature	Label	Twitch	Arxiv
GNNSafe	87.52	93.44	92.80	99.58	98.55	97.35	99.60	99.64	97.23	66.82	71.06
GNNSafe++	90.62	95.56	92.75	**99.82**	99.64	97.51	**99.99**	**99.97**	97.89	95.36	**74.77**
Graph-EBM	61.14	72.42	92.69	75.22	86.49	97.24	72.75	89.28	97.91	44.43	52.80
MC-dropout	87.50	93.12	93.03	98.67	98.50	96.80	99.51	99.53	97.25	68.58	66.34
Deep Ensemble	87.87	93.63	93.75	98.58	98.43	97.35	98.18	98.45	94.97	72.25	67.72
GPN	77.47	85.88	90.34	97.17	87.91	92.72	34.67	72.56	83.65	51.73	OOM
DSS-Hybrid	**94.32**	**97.60**	**94.11**	99.69	**99.66**	**97.52**	99.96	99.95	**98.07**	**95.75**	73.36

Table 3: Cross-graph OOD (AUROC, AUPR: ↑; FPR95, false positive rate at 95% true positive rate: ↓; ID: in-distribution; bold: best). GNNSafe/GNNSafe++ and GPN values as reported by Wu et al. [36] (OOM: out of memory on a 24 GB GPU); Graph-EBM [9], MC-dropout, and Deep Ensembles run in our pipeline (3 seeds).

	Twitch				Arxiv			
Model	AUROC	AUPR	FPR95	ID Acc.	AUROC	AUPR	FPR95	ID Acc.
GNNSafe	66.82	70.97	76.24	70.40	71.06	80.44	87.01	53.39
GNNSafe++	95.36	97.12	33.57	70.18	**74.77**	**83.21**	77.43	53.50
Graph-EBM	44.43	57.91	94.84	63.04	52.80	65.88	97.40	53.45
MC-dropout	68.58	81.25	94.84	64.10	66.34	74.88	89.69	53.56
Deep Ensemble	72.25	83.19	94.29	66.94	67.72	75.98	87.32	47.58
GPN	51.73	66.36	95.51	68.09	OOM	OOM	OOM	OOM
DSS-Hybrid	**95.75**	**97.51**	**22.40**	**70.52**	73.36	79.11	**76.67**	**53.99**

Detection comes from the energy readout, with propagation, acting on the DSS mean logit (9), which Proposition 2 formalizes as the readout that retains common-evidence shifts. Under this identical readout the DSS representation separates ID from OOD more widely than the GCN encoder alone (the Cora columns of Table 2), and a controlled ablation that varies only the chaos order $P \in \{0, \ldots, 3\}$ at one fixed configuration leaves energy AUROC flat (Cora-structure $91.0/91.2/90.9/90.8$; the level differs from Table 2 as the ablation is a separate set of runs; Appendix E), as the design predicts since the energy score reads the mean logit alone. Score propagation contributes up to $+18.2$ AUROC (Twitch) and $+14.5$ (Cora-structure) at identical checkpoints (Appendix E). Figure 2 shows the resulting wider energy-density gap between ID and OOD modes relative to the energy-based baselines.

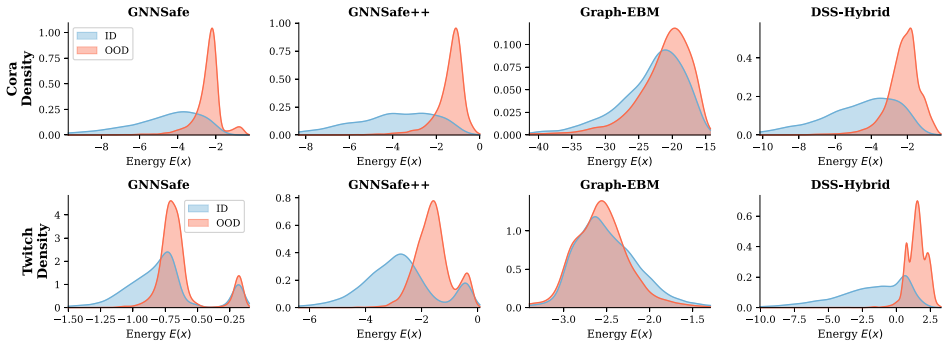

Figure 2: Energy density for ID (blue) vs. OOD (red). Top: Cora/structure. Bottom: Twitch (cross-graph). Lower energy indicates higher confidence. DSS-Hybrid separates the ID and OOD modes on both datasets.

Table 4: Test accuracy (%) on the concept-shift GOOD node benchmark for DSS-Hybrid (standard ERM training) and 13 baselines; ERM through TAR as reported by Zheng et al. [38], G-ΔUQ run in our pipeline. Best in bold, second-best underlined; OOM: out of memory.

Model	GOOD-CBAS color	GOOD-WebKB university	GOOD-Twitch language	GOOD-Cora word	GOOD-Cora degree	GOOD-Arxiv time	GOOD-Arxiv degree
ERM	82.43	27.16	51.59	64.03	60.30	65.64	54.81
IRM	82.00	26.06	49.78	63.93	60.26	65.54	56.72
VREx	82.86	26.61	55.75	64.03	60.53	65.92	56.68
Coral	81.57	28.07	51.80	64.04	60.30	65.79	55.14
DANN	83.57	29.36	51.67	63.96	60.23	65.67	55.34
SRGNN	82.14	26.42	51.58	63.96	60.27	65.64	55.08
EERM	65.71	29.91	OOM	63.42	60.21	OOM	OOM
FLOOD	84.29	28.62	54.22	64.01	60.31	65.66	58.59
CIT	83.71	28.99	OOM	63.77	60.05	OOM	OOM
KL-DRO	81.14	29.54	51.87	64.03	60.52	65.51	54.70
GroupDRO	82.71	29.17	52.24	64.10	60.43	65.93	56.24
TAR	<u>87.29</u>	<u>30.83</u>	<u>57.20</u>	<u>64.73</u>	<u>61.73</u>	<u>66.08</u>	59.26
G-ΔUQ	67.14	30.77	54.97	58.90	60.06	65.46	<u>64.34</u>
DSS-Hybrid	**88.57**	**40.77**	**61.21**	**64.81**	**62.73**	**66.76**	**66.43**

Table 5: Both deployment modes on all three tasks. Each cell summarizes the full per-dataset comparison in Appendix G (Tables 24, 25, and 26); the standalone OOD and GOOD cells and the hybrid calibration cell are the cross-evaluations; the other cells restate Tables 1, 2–3, and 4.

Task	Standalone DSS-GNN	DSS-Hybrid
Calibration (14 datasets)	Best Brier on all 14 and best accuracy on 13 (Table 1); stronger than the hybrid on 13 of 14	Never loses beyond noise to its own GCN base; Roman-Empire $+27.7$ accuracy, -0.33 Brier
OOD detection (11 settings)	Effective on 10 of 11: Cora $79.8/88.2/94.0$, Photo 96.5–98.1, CS 94.3–97.8, Arxiv 73.7 (margin objective); the Twitch energy score does not separate	Best published AUROC on 7 of 11 (Table 2); stronger than the standalone on 10 of 11
GOOD concept shift (7 settings)	Above every published baseline on 5 of 7, at ERM level on the other 2	Strongest on all 7 (Table 4)

4.4 Distribution-shifted classification

The GOOD benchmark [12] tests generalization *under* distribution shift. Table 4 compares DSS-Hybrid, trained with standard ERM, against the GOOD/TAR baselines and G-ΔUQ on seven concept-shift settings (covariate-shift splits in Appendix G.7): it has the best mean shifted accuracy in every case, by margins from 0.08 (GOOD-Cora/word) to 9.94 points (GOOD-WebKB/university). This is consistent with the inductive bias of Section 3.4: the regularized low-/high-pass residual can model the shift in which graph frequencies are predictive while the base predictor is preserved. Section 4.5 isolates the residual pathway as a unit (hybrid versus its identically trained GCN base), and Appendix E (Table 15) isolates the chaos order under shift: at the validation-selected standalone configurations, $P{=}1$ improves shifted accuracy over $P{=}0$ on GOOD-Cora/word ($+1.7$), GOOD-Cora/degree ($+1.2$), and GOOD-Arxiv/time ($+0.6$), so on those settings the *uncertainty* representation, not only the dual-filter structure, contributes.

4.5 Standalone versus hybrid: cross-evaluation and deployment guidance

Each mode is also evaluated on the other's tasks under the same protocols. Table 5 summarizes the outcome, and Tables 24–26 in Appendix G give the per-dataset numbers with standard deviations and ID accuracies (protocols in Appendix G.1).

Classification and calibration. The standalone is the stronger calibrated classifier on 13 of 14 datasets; Tolokers is the one exception (hybrid 81.2 accuracy / 0.255 Brier versus 79.9 / 0.268). Where the GCN encoder is strong the two are close (Cora 86.6 vs 84.3); on heterophilous graphs the DSS residual compensates for a weak encoder (Roman-Empire 78.8 vs 74.4); and where a plain GCN is unsuitable the standalone is higher (Texas 91.3 vs 45.6, Squirrel 68.1 vs 26.4). Against its own identically trained GCN base, the hybrid never loses beyond noise on either metric (worst

accuracy delta -0.8 on Texas, inside one standard deviation of 7.7) and gains where the encoder is weak (Roman-Empire $+27.7$ accuracy / -0.33 Brier, Minesweeper $+6.6$ / -0.11): the residual never measurably reduces its encoder's calibration or clean accuracy, and where the base is unsuitable (small heterophilous graphs) the hybrid matches it rather than improving on it.

OOD detection. The hybrid is the stronger detector on 10 of 11 settings. Its GCN-plus-residual encoder is stronger in the sparse-label perturbation regime (Cora-structure 94.3 vs 79.8) and for cross-graph transfer (Twitch), while on Arxiv the standalone run under the GNNSafe++ margin objective is 0.4 higher with the best Arxiv ID accuracy (61.2 vs 54.0). The one standalone exception is Twitch: the classifier trains (ID accuracy 68.4 vs 70.5) but its energy score does not separate the cross-graph OOD inputs, so the hybrid (95.8) is preferred there. On the perturbation settings with tiny public splits the GNNSafe pipeline gives every backbone BatchNorm, and plain GCN in the same pipeline does not train without it (ID accuracy 75.7 with BatchNorm, 45.8 without, on Cora-structure); the standalone uses one BatchNorm1d per chaos channel (Appendix D) in all OOD and GOOD cells but OOD Twitch and GOOD-Arxiv/degree, and does not need it under the full-label calibration protocol (Appendix G.6).

Distribution shift. On GOOD the standalone is above every baseline of Table 4 on 5 of 7 settings (WebKB 36.4 and Twitch 60.3 vs TAR 30.8 and 57.2) and at ERM's level on the other two (CBAS 82.4, Arxiv/time 65.4), with ID accuracy between 63.5 and 96.2 (Appendix G); the hybrid is above every baseline on all 7.

Deployment guidance. The two modes share one uncertainty representation and the same two readouts, (8) and (9), and differ in the encoder. Outside the exceptions above, each is the better mode on its deployment task and remains effective on the other's. We recommend the standalone for calibration-critical single-graph use, in particular under heterophily, and the hybrid when a strong conventional encoder exists, for cross-graph transfer, and for OOD detection in sparse-label regimes; the standalone requires BatchNorm stabilization in sparse-label regimes, and the hybrid shares its encoder's failures (Section 5).

5 Conclusion and Limitations

DSS-GNN is one uncertainty representation, a doubly-spectral stochastic expansion over graph-frequency and chaos-order axes, with task-matched readouts and two deployment modes. Empirically, the standalone has the lowest Brier score among the compared uncertainty-aware baselines on all 14 calibration benchmarks, the hybrid the strongest AUROC on most node-OOD settings and the strongest shifted accuracy on all 7 concept-shift benchmarks under ERM, and both modes are effective on all three tasks with the exceptions noted in Section 4.5.

Limitations. (i) *Two deployment modes.* The modes share one representation and the same readouts but are not interchangeable in every regime. The standalone model requires BatchNorm stabilization in sparse-label regimes, where every baseline backbone in the OOD pipeline also uses it, and uses it on 6 of 7 GOOD settings; under the full-label calibration protocol it is not needed and is not used (Appendix G.6). The hybrid is preferable when a strong conventional encoder exists and shares that encoder's limits: on small heterophilous graphs the hybrid matches its base without improving on it. The standalone energy score does not separate cross-graph OOD inputs on Twitch, where only the hybrid is effective. (ii) *Scalar latent.* All stochastic variation is due to one shared Gaussian factor, perfectly dependent across nodes, so the logit covariance across nodes has rank at most P. This suffices for the marginal readouts used here (per-node calibration and energy scores) and is what makes single-pass quadrature possible, but it cannot represent independent per-node noise, so joint cross-node uncertainty is not modeled; the sampling baselines (MC-dropout, Deep Ensembles) have higher Brier than the standalone and lower AUROC than the hybrid (Appendix F, Table 2). A vector latent of dimension q would cost S^q quadrature nodes on a full tensor grid, which sparse grids mitigate; we leave this to future work. (iii) *Compute.* The chaos expansion adds a $(P{+}1) \times S$ operation count per layer; the measured per-epoch overhead at $P{=}2$ is $2.9\times$ on Cora and $1.2\times$ on Amazon-Ratings, and about 4% per additional order on ogbn-arxiv, with memory linear in $P{+}1$ (Table 20).

Acknowledgments and Disclosure of Funding

This work was partially supported by NSF 2303037, NSF 2312501, NSF 2531008, NSF 2553619, SRC JUMP 2.0 Center, CDSC Industry Partners, Amazon Research Awards, Snapchat, and Google Gifts. Part of this work was done while Fred Xu was an intern at Block, Inc., and we thank Block, Inc. for its support.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. doi: 10.48550/arXiv.1907.02893. URL https://arxiv.org/abs/1907.02893.

[2] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35 (5):3950–3957, 2021. doi: 10.1609/aaai.v35i5.16514. URL https://ojs.aaai.org/index.php/AAAI/article/view/16514.

[3] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. URL https://doi.org/10.1175/1520-0493(1950)078%3C0001:VOFEIT%3E2.0.CO;2.

[4] Robert H. Cameron and William T. Martin. The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals. *Annals of Mathematics*, 48(2):385–392, 1947. doi: 10.2307/1969178. URL https://doi.org/10.2307/1969178.

[5] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized PageRank graph neural network. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

[7] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 315–324. ACM, 2020. doi: 10.1145/3340531.3411903. URL https://doi.org/10.1145/3340531.3411903.

[8] Rui Duan, Mingjian Guang, Junli Wang, Chungang Yan, Hongda Qi, Wenkang Su, Can Tian, and Haoran Yang. Unifying homophily and heterophily for spectral graph neural networks via triple filter ensembles. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-2966. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a9db2b121c3517fd559ecbe5038701ee-Abstract-Conference.html.

[9] Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. Energy-based epistemic uncertainty for graph neural networks. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-1084. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3cd50f2922b7adaaa9e5113e35bae095-Abstract-Conference.html.

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

[11] Gene H. Golub and John H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969. doi: 10.1090/S0025-5718-69-99647-1. URL https://doi.org/10.1090/S0025-5718-69-99647-1.

[12] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. GOOD: A graph out-of-distribution benchmark. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0dc91de822b71c66a7f54fa121d8cbb9-Abstract-Datasets_and_Benchmarks.html.

[13] Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/76f1cfd7754a6e4fc3281bcccb3d0902-Abstract.html.

[14] Svante Janson. *Gaussian Hilbert Spaces*. Cambridge University Press, 1997. doi: 10.1017/CBO9780511526169. URL https://doi.org/10.1017/CBO9780511526169.

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

[16] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (REx). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5815–5826. PMLR, 2021. URL https://proceedings.mlr.press/v139/krueger21a.html.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

[19] Xixun Lin, Wenxiao Zhang, Fengzhao Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, Shirui Pan, and Yanan Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30457–30478. PMLR, 2024. URL https://proceedings.mlr.press/v235/lin24x.html.

[20] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html.

[21] Edward Nelson. The free Markoff field. *Journal of Functional Analysis*, 12(2):211–227, 1973. doi: 10.1016/0022-1236(73)90025-6. URL https://doi.org/10.1016/0022-1236(73)90025-6.

[22] David Nualart. *The Malliavin Calculus and Related Topics*. Probability and Its Applications. Springer Berlin Heidelberg, 2nd edition, 2006. doi: 10.1007/3-540-28329-3. URL https://doi.org/10.1007/3-540-28329-3.

[23] Sergey Oladyshkin, Timothy Praditia, Ilja Kröker, Farid Mohammadi, Wolfgang Nowak, and Sebastian Otte. The deep arbitrary polynomial chaos neural network or how Deep Artificial Neural Networks could benefit from data-driven homogeneous chaos theory. *Neural Networks*, 166:85–104, 2023. doi: 10.1016/j.neunet.2023.06.036. URL https://doi.org/10.1016/j.neunet.2023.06.036.

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

[25] 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. URL https://openreview.net/forum?id=S1e2agrFvS.

[26] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

[28] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS 2018*, 2018. doi: 10.48550/arXiv.1811.05868. URL https://arxiv.org/abs/1811.05868.

[29] 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. URL https://doi.org/10.1109/MSP.2012.2235192.

[30] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. Graph Posterior Network: Bayesian predictive uncertainty for node classification. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/95b431e51fc53692913da5263c214162-Abstract.html.

[31] Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, 2020. doi: 10.1016/j.cell.2020.01.021. URL https://doi.org/10.1016/j.cell.2020.01.021.

[32] Gábor Szegő. *Orthogonal Polynomials*. American Mathematical Society, 4th edition, 1975.

[33] Puja Trivedi, Mark Heimann, Rushil Anirudh, Danai Koutra, and Jayaraman J. Thiagarajan. Accurate and scalable estimation of epistemic uncertainty for graph neural networks. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZL6yd6N1S2.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

[35] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. doi: 10.2307/2371268. URL https://doi.org/10.2307/2371268.

[36] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zoz7Ze4STUL.

[37] Dongbin Xiu and George Em Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002. doi: 10.1137/S1064827501387826. URL https://doi.org/10.1137/S1064827501387826.

[38] Weihuang Zheng, Jiashuo Liu, Jiaxing Li, Jiayun Wu, Peng Cui, and Youyong Kong. Topology-aware dynamic reweighting for distribution shifts on graph. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 78260–78275. PMLR, 2025. URL https://proceedings.mlr.press/v267/zheng25k.html.

Appendix: Table of Contents

A Proofs and Derivations **16**

A.1 Background: Wiener chaos and Gauss–Hermite quadrature 16

A.2 Supporting technical results 18

A.3 Proof of Proposition 1 (Generalization) 19

A.4 Proof of Theorem 1 (Doubly-spectral operator) 19

A.5 Proof of Theorem 2 (Representational capacity and truncation) 21

A.6 Proof of the Truncation Bound in Theorem 2 23

A.7 Proof of Proposition 2 (Stochastic and mean-logit readouts) 24

B Baseline Details **27**

B.1 Baseline methods 27

C Dataset Statistics and Licenses **28**

C.1 Calibration benchmarks 28

C.2 OOD detection benchmarks 29

C.3 GOOD distribution-shift benchmarks 29

D Implementation Details **29**

D.1 General training protocol 29

D.2 Calibration (Table 1) 29

D.3 OOD detection (Section 4.3) 30

D.4 Distribution shift (Table 4) 31

D.5 Software, hardware, and compute budget 31

E Ablation Study Details **32**

E.1 Polynomial order P across all datasets 32

E.2 Chaos order on OOD detection and under distribution shift 32

E.3 Energy-score propagation 33

E.4 Chaos energy regularization λ_{reg} 33

E.5 Number of quadrature nodes S 34

E.6 Contribution decomposition: backbone vs. chaos 35

E.7 Computational overhead 35

E.8 Summary of component evidence 36

F Comparison with Standard GNN Baselines **37**

G Cross-Evaluation of the Two Deployment Modes **39**

G.1 Protocols 39

G.2 Calibration: standalone, hybrid, and the hybrid's GCN base 39

G.3 OOD detection: standalone versus hybrid 40

G.4 Distribution shift: standalone versus hybrid 40
G.5 Readout coincidence 41
G.6 BatchNorm under the calibration protocol 41
G.7 Covariate shift on GOOD 41

H Architecture Overview and Notation **43**

A Proofs and Derivations

We begin with a self-contained introduction to Wiener chaos and Gauss–Hermite quadrature (Appendix A.1), then present supporting technical results (Appendix A.2), followed by individual proofs of each main-text theorem (Appendices A.3–A.7).

A.1 Background: Wiener chaos and Gauss–Hermite quadrature

This section collects the probability and approximation theory underlying the doubly-spectral expansion. The material is classical; we follow the presentation in Janson [14] and Xiu and Karniadakis [37]. This appendix is self-contained.

Gaussian measure and Hermite polynomials. Let γ denote the standard Gaussian measure on $\mathbb{R}$, i.e., $d\gamma(\omega) = (2\pi)^{-1/2} e^{-\omega^2/2}\, d\omega$. The Hilbert space $L^2(\mathbb{R}, \gamma)$ consists of all square-integrable functions with respect to γ, equipped with the inner product $\langle f, g\rangle = \int_{\mathbb{R}} f(\omega)\, g(\omega)\, d\gamma(\omega) = \mathbb{E}[f(\omega)\, g(\omega)]$ where $\omega \sim \mathcal{N}(0, 1)$.

The *probabilists' Hermite polynomials* [32] are defined by the Rodrigues formula

$$\mathrm{He}_n(\omega) = (-1)^n\, e^{\omega^2/2}\, \frac{d^n}{d\omega^n}\, e^{-\omega^2/2}, \qquad n = 0, 1, 2, \ldots \tag{12}$$

The first few are $\mathrm{He}_0(\omega) = 1$, $\mathrm{He}_1(\omega) = \omega$, $\mathrm{He}_2(\omega) = \omega^2 - 1$, $\mathrm{He}_3(\omega) = \omega^3 - 3\omega$. They satisfy the three-term recurrence

$$\mathrm{He}_{n+1}(\omega) = \omega\, \mathrm{He}_n(\omega) - n\, \mathrm{He}_{n-1}(\omega), \tag{13}$$

with $\mathrm{He}_{-1} = 0$, and the orthogonality relation $\mathbb{E}[\mathrm{He}_m(\omega)\, \mathrm{He}_n(\omega)] = n!\, \delta_{mn}$. Normalizing gives the *orthonormal Hermite basis*

$$\Psi_n(\omega) := \frac{\mathrm{He}_n(\omega)}{\sqrt{n!}}, \qquad \mathbb{E}[\Psi_m \Psi_n] = \delta_{mn}, \tag{14}$$

which forms a complete orthonormal system for $L^2(\mathbb{R}, \gamma)$ [14, 32]. Every $f \in L^2(\mathbb{R}, \gamma)$ can therefore be expanded as $f(\omega) = \sum_{n=0}^{\infty} f_n\, \Psi_n(\omega)$ with $f_n = \mathbb{E}[f \Psi_n]$ and Parseval's identity $\|f\|^2_{L^2(\gamma)} = \sum_n f_n^2$.

Wiener–Itô chaos decomposition. The Hermite expansion generalizes to a fundamental structural result in Gaussian analysis. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space carrying a standard Gaussian random variable ω, and define the associated Gaussian subspace

$$L^2_\omega := \{f(\omega) : f \in L^2(\mathbb{R}, \gamma)\} \subseteq L^2(\Omega, \mathbb{P}).$$

Define the n-th *Wiener chaos* $\mathcal{C}_n$ as the closed linear span of $\Psi_n(\omega)$ in L^2_ω. Since we use a single Gaussian latent ($q = 1$), each $\mathcal{C}_n$ is one-dimensional and spanned by $\Psi_n(\omega)$; with $q > 1$, the corresponding chaos subspace would be indexed by Hermite multi-indices. The *Wiener–Itô chaos decomposition* states [4, 14, 35]:

$$L^2_\omega = \bigoplus_{n=0}^{\infty} \mathcal{C}_n, \tag{15}$$

that is, every square-integrable random variable measurable with respect to ω admits a unique, orthogonal expansion in the Hermite basis. Denoting the orthogonal projection onto $\mathcal{C}_n$ by P_n, any $f \in L^2_\omega$ satisfies $f = \sum_{n=0}^{\infty} P_n f$ with $\|f\|^2 = \sum_n \|P_n f\|^2$. The zeroth chaos $\mathcal{C}_0$ consists of constants ($P_0 f = \mathbb{E}[f]$); the first chaos is the space of centered Gaussian random variables; higher chaoses capture progressively more complex nonlinear functionals of ω.

Remark 1 (Analogy with graph spectral decomposition)**.** *The decomposition* (15) *is the stochastic counterpart of the graph spectral decomposition* $\ell^2(V) = \bigoplus_{j=1}^{N} \mathrm{span}(u_j)$. *Both decompose a Hilbert space into eigenspaces of a self-adjoint operator: the graph Laplacian* L_G *for* $\ell^2(V)$*, and the Ornstein–Uhlenbeck operator* $\mathcal{L}$ *(defined below) for the Gaussian Hilbert space* L^2_ω.

Remark 2 (Completeness of the doubly-spectral basis)**.** *Take the signal domain to be the random graph signals generated by the shared latent: a random variable per node with finite second moment, measurable with respect to ω, i.e. an element of the tensor-product Hilbert space $\ell^2(V) \otimes L^2(\mathbb{R}, \gamma)$. The graph Fourier eigenvectors $\{u_j\}_{j=1}^N$ form an orthonormal basis of $\ell^2(V)$ and, by the Wiener–Itô decomposition* (15)*, the normalized Hermite polynomials $\{\Psi_n\}_{n\geq 0}$ form an orthonormal basis of $L^2(\mathbb{R}, \gamma)$. The tensor product of orthonormal bases is an orthonormal basis of the tensor-product space, so $\{u_j \otimes \Psi_n\}_{j\leq N,\, n\geq 0}$ is a complete orthonormal basis of this domain, and the double expansion* (1) *is a bi-spectral decomposition in the strict sense: both factors are eigenbases of self-adjoint operators (Remark 1). The scope caveat is that the space is built over the scalar ω: signals with independent per-node latents are outside it (Section 5).*

The Ornstein–Uhlenbeck operator and hypercontractivity. The *Ornstein–Uhlenbeck (O-U) operator* $\mathcal{L}$ is defined on its standard dense domain in $L^2(\mathbb{R}, \gamma)$ by

$$\mathcal{L}f(\omega) = f''(\omega) - \omega\, f'(\omega), \tag{16}$$

with eigenvalues and eigenfunctions $\mathcal{L}\Psi_n = -n\,\Psi_n$ [22]. This is directly analogous to the graph Laplacian having eigenpairs (u_j, λ_j): $-\mathcal{L}$ is the corresponding operator on $L^2(\mathbb{R}, \gamma)$, with the chaos orders $0, 1, 2, \ldots$ as eigenvalues. The associated *O-U semigroup* $T_t = e^{t\mathcal{L}}$ acts as $T_t\Psi_n = e^{-nt}\Psi_n$, damping higher-order chaos components exponentially.

A fundamental property of T_t is *hypercontractivity*: for $1 < p \leq q$ and $e^{2t} \geq (q-1)/(p-1)$,

$$\|T_t f\|_{L^q(\gamma)} \leq \|f\|_{L^p(\gamma)} \tag{17}$$

for all $f \in L^p(\gamma)$ [14, 21]. A direct consequence for chaos components is the *moment comparison inequality*: for every $g \in \mathcal{C}_n$ and $q \geq 2$,

$$\|g\|_{L^q(\gamma)} \leq (q-1)^{n/2}\, \|g\|_{L^2(\gamma)}. \tag{18}$$

This follows by applying (17) with $p = 2$ and $\rho = e^{-t} = (q-1)^{-1/2}$ to $g \in \mathcal{C}_n$, using $T_t g = e^{-nt} g$ [14, Corollary 5.11]. Equation (18) bounds the higher moments of a fixed chaos component; the exponential *decay* of chaos coefficients with order n requires additional regularity and is established separately in Theorem 2 via a Cauchy-estimate argument.

Gauss–Hermite quadrature. To evaluate integrals against γ numerically, we use Gauss–Hermite quadrature [11, 37]. An S-point rule $\{(\omega_s, \mu_s)\}_{s=1}^S$ approximates

$$\int_{\mathbb{R}} f(\omega)\, d\gamma(\omega) \;\approx\; \sum_{s=1}^{S} \mu_s\, f(\omega_s), \tag{19}$$

where the nodes $\omega_1 < \cdots < \omega_S$ are the roots of $\mathrm{He}_S(\omega)$ and the weights $\mu_s > 0$ sum to one. The main property is *polynomial exactness*: the approximation is exact whenever f is a polynomial of degree at most $2S - 1$ [32]. In particular:

- Orthonormality: $\sum_{s=1}^S \mu_s\, \Psi_m(\omega_s)\, \Psi_n(\omega_s) = \delta_{mn}$ for all $m + n \leq 2S - 1$.
- Triple products: $\sum_{s=1}^S \mu_s\, \Psi_r(\omega_s)\Psi_n(\omega_s)\Psi_m(\omega_s) = \mathbb{E}[\Psi_r\Psi_n\Psi_m]$ whenever $r + n + m \leq 2S - 1$.

For DSS-GNN with chaos order P and gate order P_g, the most demanding integrand is the triple product $\Psi_r\Psi_n\Psi_m$ with $r \leq P_g$, $n, m \leq P$, which has degree $P_g + 2P$. Exactness therefore requires $S \geq \lceil (P_g + 2P + 1)/2 \rceil$ quadrature nodes. For $P_g \leq 1$ the threshold is $S \geq P + 1$; the experiments use $S = 4$, which is exact for $P \leq 3$.

Remark 3 (Non-intrusive vs. intrusive projection)**.** *In the polynomial chaos literature [37], two strategies compute the chaos coefficients of a transformed variable $\sigma(H(\omega))$: (i)* intrusive *methods analytically propagate the expansion through σ using triple-product coefficients (Lemma 1); (ii)* non-intrusive *methods evaluate σ at quadrature nodes and project back* (19)*. DSS-GNN uses the non-intrusive approach* (6) *because it applies unchanged to* any *pointwise activation σ without requiring closed-form triple-product expansions. Theorem 3 proves exact agreement for product terms $\alpha(\omega)H(\omega)$ when the quadrature rule integrates the relevant triple products exactly; for a general nonlinear activation σ, the non-intrusive projection remains an approximation whose error is decomposed in Proposition 3.*

A.2 Supporting technical results

Lemma 1 (Products in chaos: triple-product contraction)**.** *Let $\alpha(\omega) = \sum_{r=0}^{P_g} a_r \Psi_r(\omega)$ and $H(\omega) = \sum_{n=0}^{P} H_n \Psi_n(\omega)$. Then the order-$m$ coefficient of the product $\alpha(\omega)H(\omega)$ is*

$$(\alpha H)_m = \sum_{r=0}^{P_g} \sum_{n=0}^{P} a_r H_n \, c_{rnm}, \qquad c_{rnm} := \mathbb{E}[\Psi_r \Psi_n \Psi_m].$$

Proof. Expand the product in the Hermite basis:

$$\alpha(\omega)H(\omega) = \Big(\sum_{r=0}^{P_g} a_r \Psi_r(\omega)\Big)\Big(\sum_{n=0}^{P} H_n \Psi_n(\omega)\Big) = \sum_{r,n} a_r H_n \, \Psi_r(\omega)\Psi_n(\omega).$$

The order-m coefficient is obtained by projecting onto Ψ_m using the orthonormality $\mathbb{E}[\Psi_i \Psi_j] = \delta_{ij}$:

$$(\alpha H)_m = \mathbb{E}\big[\alpha(\omega)H(\omega)\Psi_m(\omega)\big] = \sum_{r,n} a_r H_n \, \mathbb{E}[\Psi_r \Psi_n \Psi_m] = \sum_{r,n} a_r H_n \, c_{rnm}.$$

The triple-product constants $c_{rnm} = \mathbb{E}[\Psi_r \Psi_n \Psi_m]$ are computable in closed form from the three-term recurrence of the probabilists' Hermite polynomials [37]. In particular, $c_{rnm} = 0$ unless $|r - n| \le m \le r + n$ and $r + n + m$ is even, which makes the coupling matrices sparse. □

Theorem 3 (Non-intrusive projection recovers product coefficients when quadrature is exact)**.** *Let $\widehat{(\alpha H)}_m := \sum_{s=1}^{S} \mu_s \, \alpha(\omega_s) H(\omega_s) \Psi_m(\omega_s)$ be the discrete projection of the product using a Gauss–Hermite quadrature rule $\{(\omega_s, \mu_s)\}_{s=1}^{S}$. If the quadrature integrates all triple products $\Psi_r(\omega)\Psi_n(\omega)\Psi_m(\omega)$ exactly for $r \le P_g$, $n \le P$, $m \le P$, then $\widehat{(\alpha H)}_m = (\alpha H)_m$ for all $m \le P$.*

Proof. Substituting the expansions of α and H:

$$\begin{aligned}
\widehat{(\alpha H)}_m &= \sum_{s=1}^{S} \mu_s \, \alpha(\omega_s) H(\omega_s) \Psi_m(\omega_s) \\
&= \sum_{s=1}^{S} \mu_s \sum_{r=0}^{P_g} \sum_{n=0}^{P} a_r H_n \, \Psi_r(\omega_s)\Psi_n(\omega_s)\Psi_m(\omega_s) \\
&= \sum_{r,n} a_r H_n \underbrace{\sum_{s=1}^{S} \mu_s \, \Psi_r(\omega_s)\Psi_n(\omega_s)\Psi_m(\omega_s)}_{Q_S[\Psi_r \Psi_n \Psi_m]}.
\end{aligned}$$

The integrand $\Psi_r \Psi_n \Psi_m$ is a polynomial of degree $r + n + m \le P_g + 2P$ under the Gaussian weight γ. An S-point Gauss–Hermite rule integrates polynomials of degree up to $2S - 1$ exactly [37]. The condition $2S - 1 \ge P_g + 2P$, i.e. $S \ge \lceil (P_g + 2P + 1)/2 \rceil$, therefore ensures $Q_S[\Psi_r \Psi_n \Psi_m] = \mathbb{E}[\Psi_r \Psi_n \Psi_m] = c_{rnm}$. By Lemma 1, $\widehat{(\alpha H)}_m = (\alpha H)_m$. □

Proposition 3 (Discrete projection error decomposition)**.** *Let Π_P be the exact $L^2(\mathbb{R}, \gamma)$ projection onto the first $P + 1$ Wiener chaos orders, and let $\Pi_{P,S} Q$ be the finite chaos expansion whose coefficients are induced by a quadrature rule: $(\Pi_{P,S} Q)_m := \sum_{s=1}^{S} \mu_s \, Q(\omega_s) \Psi_m(\omega_s)$. Then for any $Q \in L^2(\mathbb{R}, \gamma)$,*

$$\|Q - \Pi_{P,S} Q\|_{L^2} \;\le\; \underbrace{\|Q - \Pi_P Q\|_{L^2}}_{\textit{truncation error}} + \underbrace{\|\Pi_P Q - \Pi_{P,S} Q\|_{L^2}}_{\textit{quadrature error}}.$$

Moreover, Gauss–Hermite rules yield exact coefficients whenever $Q(\omega)\Psi_m(\omega)$ is a polynomial of sufficiently low degree under the Gaussian weight.

Proof. The triangle inequality in $L^2(\mathbb{R},\gamma)$ gives $\|Q-\Pi_{P,S}Q\| \le \|Q-\Pi_P Q\| + \|\Pi_P Q-\Pi_{P,S}Q\|$. For the truncation term, Π_P is the orthogonal projection onto $\bigoplus_{n=0}^{P}\mathcal{C}_n$, the direct sum of the first $P+1$ Wiener chaos subspaces [14], so $\|Q-\Pi_P Q\|^2 = \sum_{n>P}\|Q_n\|^2$ where $Q_n = \mathbb{E}[Q\Psi_n]$ are the exact chaos coefficients. For the quadrature term, the m-th coefficient of $\Pi_{P,S}Q$ is $\sum_s \mu_s Q(\omega_s)\Psi_m(\omega_s)$, which approximates $\int Q\Psi_m\, d\gamma$. When $Q\Psi_m$ is a polynomial of degree at most $2S-1$, Gauss–Hermite quadrature integrates it exactly [37], making the quadrature error zero. In the DSS layer context, the pre-activation $\widetilde{H}^{(\ell+1)}(\omega)$ before applying σ is a polynomial of degree at most P_g+P in ω; after applying σ it is no longer polynomial, and the quadrature error reflects the approximation quality of the S-point rule on this non-polynomial integrand. □

A.3 Proof of Proposition 1 (Generalization)

Proposition 1 (Generalization)**.** *With $P=0$ and $P_g=0$,* (2) *has a single deterministic coefficient $H_0^{(\ell)}$ with scalar gates, and the DSS layer reduces to a standard polynomial spectral GNN layer, recovering ChebNet, GCN, and related models as special cases.*

Proof. When $P=0$, the chaos expansion (2) contains a single term: $H^{(\ell)}(\omega) = H_0^{(\ell)}\Psi_0(\omega)$. Since the zeroth normalized Hermite polynomial satisfies $\Psi_0 \equiv 1$ [37], the representation is deterministic: $H^{(\ell)}(\omega) = H_0^{(\ell)}$ for all $\omega \in \Omega$.

Step 1 (Gate reduction). With $P=0$, the mixing weights (4) are indexed by $n=0$ only. Setting the gate order $P_g=0$ (no stochastic gating when the representation itself is deterministic) yields scalar constants $\alpha_0 = a_{0,0} \in \mathbb{R}$ and $\beta_0 = b_{0,0} \in \mathbb{R}$.

Step 2 (Pre-activation collapse). The quadrature-node pre-activation (5) reduces to

$$\begin{aligned}\widetilde{H}^{(\ell+1)}(\omega_s) &= \Psi_0(\omega_s)\big(\alpha_0\, G_{\mathrm{lp}}(L_G)H_0^{(\ell)} + \beta_0\, G_{\mathrm{hp}}(L_G)H_0^{(\ell)}\big)W_\ell \\ &= \big(\alpha_0\, G_{\mathrm{lp}}(L_G) + \beta_0\, G_{\mathrm{hp}}(L_G)\big)H_0^{(\ell)}W_\ell,\end{aligned}$$

which is independent of the quadrature node ω_s.

Step 3 (Projection collapse). The projection (6) for $m=0$ becomes

$$H_0^{(\ell+1)} = \sum_{s=1}^{S}\mu_s\,\sigma\big(\widetilde{H}^{(\ell+1)}(\omega_s)\big)\Psi_0(\omega_s) = \sigma\big(\widetilde{H}^{(\ell+1)}\big)\sum_{s=1}^{S}\mu_s = \sigma\big(\widetilde{H}^{(\ell+1)}\big),$$

where the second equality uses the fact that $\widetilde{H}^{(\ell+1)}$ does not depend on ω_s (Step 2), and the third uses the normalization $\sum_s \mu_s = \int_{\mathbb{R}} d\gamma = 1$ of the Gauss–Hermite weights [37].

Step 4 (Recovery of spectral GNN). Combining Steps 2 and 3, the full DSS layer update at $P=0$ is

$$H_0^{(\ell+1)} = \sigma\big(g(L_G)\,H_0^{(\ell)}\,W_\ell\big), \qquad g(L_G) := \alpha_0\, G_{\mathrm{lp}}(L_G) + \beta_0\, G_{\mathrm{hp}}(L_G).$$

Since G_{lp} and G_{hp} are Chebyshev polynomials in L_G (3), their linear combination $g(L_G)$ is a polynomial in L_G of degree $\max(K_{\mathrm{lp}}, K_{\mathrm{hp}})$. This is exactly the form of a polynomial spectral graph convolution layer [6, 29]: the filter g acts in the graph-frequency domain as $g(\lambda_j) = \alpha_0\, g_{\mathrm{lp}}(\lambda_j) + \beta_0\, g_{\mathrm{hp}}(\lambda_j)$, where $g_{\mathrm{lp}}(\lambda) = \sum_k c_k^{(\mathrm{lp})} T_k(\tilde{\lambda})$ is the scalar frequency response of the low-pass branch and $\tilde{\lambda} = 2\lambda/\lambda_{\max} - 1$.

Special cases. Setting $\beta_0 = 0$ and $K_{\mathrm{lp}} = 1$ with the renormalization trick of Kipf and Welling [15] yields GCN as a single-layer DSS-GNN with $P=0$. Allowing arbitrary Chebyshev degree with both filter branches active recovers ChebNet [6] in its scalar-coefficient form (filter weights $\Theta_k = c_k W_\ell$). With generalized polynomial coefficient learning, GPR-GNN [5] and BernNet [13] are similarly obtained as $P=0$ specializations. □

A.4 Proof of Theorem 1 (Doubly-spectral operator)

Lemma 2 (Exact-projection conditions for the joint transfer)**.** *In Theorem 1, assume the pointwise nonlinearity is omitted, the gates have chaos degree at most P_g, the state has chaos degree at most P, and the Gauss–Hermite rule integrates all triple products $\Psi_r\Psi_n\Psi_m$ with $r \le P_g$ and $n, m \le P$.*

A sufficient condition is $S \geq \lceil (P_g + 2P + 1)/2 \rceil$. Under this condition, the stochastic coupling matrices in the joint-transfer formula are

$$M^{(\ell,\mathrm{lp})}_{mn} := \sum_{r=0}^{P_g} a_{n,r}\, c_{rnm}, \qquad M^{(\ell,\mathrm{hp})}_{mn} := \sum_{r=0}^{P_g} b_{n,r}\, c_{rnm}, \qquad c_{rnm} := \mathbb{E}[\Psi_r \Psi_n \Psi_m].$$

Proof. The product $\Psi_r \Psi_n \Psi_m$ has degree at most $P_g + 2P$ under the stated index ranges. An S-point Gauss–Hermite rule is exact for polynomials of degree at most $2S-1$, so $2S-1 \geq P_g + 2P$ suffices. The coupling matrix definitions are obtained by collecting the exactly projected triple-product coefficients in the linearized DSS update. □

Theorem 1 (Doubly-spectral operator)**.** *With the pointwise nonlinearity omitted, gates of chaos degree at most P_g, states of degree at most P, and a Gauss–Hermite rule exact for all triple products $\Psi_r \Psi_n \Psi_m$ ($r \leq P_g$; $n, m \leq P$; $S \geq \lceil (P_g + 2P + 1)/2 \rceil$ suffices), the linearized DSS layer is a linear operator on the joint graph-frequency/chaos-order domain of* (1)*, acting frequency by frequency: the Chebyshev filters set the graph-frequency response and the chaos gates the chaos-order mixing (Appendix A.4; the omitted nonlinearity's projection error is decomposed in Proposition 3).*

Proof. Let $\{u_j\}_{j=1}^N$ be the orthonormal eigenvectors of the graph Laplacian L_G with eigenvalues $\{\lambda_j\}_{j=1}^N$, so that $L_G = \sum_{j=1}^N \lambda_j\, u_j u_j^\top$ [29]. Because the Chebyshev filters (3) are polynomials in L_G, each eigenvector is simultaneously an eigenvector of both filters:

$$G_{\mathrm{lp}}(L_G)\, u_j = g_{\mathrm{lp}}(\lambda_j)\, u_j, \qquad G_{\mathrm{hp}}(L_G)\, u_j = g_{\mathrm{hp}}(\lambda_j)\, u_j, \tag{20}$$

where $g_{\mathrm{lp}}(\lambda) = \sum_k c_k^{(\mathrm{lp})} T_k(\tilde{\lambda})$ and $g_{\mathrm{hp}}(\lambda) = \sum_k c_k^{(\mathrm{hp})} T_k(\tilde{\lambda})$ are the scalar frequency responses.

Step 1 (Graph Fourier transform of chaos coefficients). Define the joint spectral coefficient at graph frequency λ_j and chaos order n as $\hat{H}_n^{(\ell)}(\lambda_j) := u_j^\top H_n^{(\ell)} \in \mathbb{R}^{1 \times d_\ell}$. For proof convenience, write $U_n^{\mathrm{lp}} := G_{\mathrm{lp}}(L_G) H_n^{(\ell)}$ and $U_n^{\mathrm{hp}} := G_{\mathrm{hp}}(L_G) H_n^{(\ell)}$. These spectrally filtered coefficients from (5) satisfy

$$u_j^\top U_n^{\mathrm{lp}} = u_j^\top G_{\mathrm{lp}}(L_G)\, H_n^{(\ell)} = g_{\mathrm{lp}}(\lambda_j)\, \hat{H}_n^{(\ell)}(\lambda_j), \tag{21}$$

and analogously $u_j^\top U_n^{\mathrm{hp}} = g_{\mathrm{hp}}(\lambda_j)\, \hat{H}_n^{(\ell)}(\lambda_j)$.

Step 2 (Linearized update in the chaos basis). Omitting the pointwise nonlinearity σ (linearization), the DSS update (5)–(6) gives

$$\begin{aligned} H_m^{(\ell+1)} &= \sum_{s=1}^{S} \mu_s\, \widetilde{H}^{(\ell+1)}(\omega_s)\, \Psi_m(\omega_s) \\ &= \sum_{s=1}^{S} \mu_s \sum_{n=0}^{P} \Psi_n(\omega_s) \big(\alpha_n(\omega_s)\, U_n^{\mathrm{lp}} + \beta_n(\omega_s)\, U_n^{\mathrm{hp}}\big) W_\ell\, \Psi_m(\omega_s). \end{aligned} \tag{22}$$

Expanding the gates $\alpha_n(\omega_s) = \sum_{r=0}^{P_g} a_{n,r} \Psi_r(\omega_s)$ from (4) and exchanging the order of summation:

$$\begin{aligned} H_m^{(\ell+1)} = \sum_{n=0}^{P} \Big[\Big(\sum_{r=0}^{P_g} a_{n,r} \underbrace{\sum_{s=1}^{S} \mu_s\, \Psi_r(\omega_s) \Psi_n(\omega_s) \Psi_m(\omega_s)}_{= c_{rnm}} \Big) U_n^{\mathrm{lp}} \\ + \Big(\sum_{r=0}^{P_g} b_{n,r}\, c_{rnm} \Big) U_n^{\mathrm{hp}} \Big] W_\ell, \end{aligned} \tag{23}$$

where the identification $\sum_s \mu_s\, \Psi_r(\omega_s) \Psi_n(\omega_s) \Psi_m(\omega_s) = c_{rnm} := \mathbb{E}[\Psi_r \Psi_n \Psi_m]$ follows from Theorem 3: the Gauss–Hermite rule integrates the triple product exactly when S is sufficiently large.

Define the *chaos coupling matrices* $M^{(\ell,\mathrm{lp})},\, M^{(\ell,\mathrm{hp})} \in \mathbb{R}^{(P+1)\times(P+1)}$ with entries

$$M^{(\ell,\mathrm{lp})}_{mn} := \sum_{r=0}^{P_g} a_{n,r}\, c_{rnm}, \qquad M^{(\ell,\mathrm{hp})}_{mn} := \sum_{r=0}^{P_g} b_{n,r}\, c_{rnm}. \tag{24}$$

Then (23) simplifies to

$$H_m^{(\ell+1)} = \sum_{n=0}^{P} \big(M_{mn}^{(\ell,\mathrm{lp})}\, U_n^{\mathrm{lp}} + M_{mn}^{(\ell,\mathrm{hp})}\, U_n^{\mathrm{hp}} \big)\, W_\ell. \tag{25}$$

Step 3 (Joint spectral representation). Left-multiplying (25) by $u_j^\top$ and applying (21):

$$\hat{H}_m^{(\ell+1)}(\lambda_j) = \sum_{n=0}^{P} \big(M_{mn}^{(\ell,\mathrm{lp})}\, g_{\mathrm{lp}}(\lambda_j) + M_{mn}^{(\ell,\mathrm{hp})}\, g_{\mathrm{hp}}(\lambda_j) \big)\, \hat{H}_n^{(\ell)}(\lambda_j)\, W_\ell.$$

Stack the chaos orders into a matrix $\widehat{\mathbf{H}}^{(\ell)}(\lambda_j) := \big[\hat{H}_0^{(\ell)}(\lambda_j);\ \dots\ ;\ \hat{H}_P^{(\ell)}(\lambda_j)\big] \in \mathbb{R}^{(P+1)\times d_\ell}$. The equation above is the $(m,\cdot)$ row of the matrix relation

$$\widehat{\mathbf{H}}^{(\ell+1)}(\lambda_j) = \mathcal{H}^{(\ell)}(\lambda_j)\, \widehat{\mathbf{H}}^{(\ell)}(\lambda_j)\, W_\ell, \tag{26}$$

where the $(P{+}1) \times (P{+}1)$ transfer matrix decomposes as

$$\mathcal{H}^{(\ell)}(\lambda) = g_{\mathrm{lp}}(\lambda)\, M^{(\ell,\mathrm{lp})} + g_{\mathrm{hp}}(\lambda)\, M^{(\ell,\mathrm{hp})}. \tag{27}$$

Interpretation. Equation (26) shows that, in the linearized regime, the DSS layer implements a *learned linear operator on the joint spectral domain* of the doubly-spectral expansion (1). The decomposition (27) separates two independent mechanisms:

1. *Graph-frequency response*: the scalar functions $g_{\mathrm{lp}}(\lambda)$ and $g_{\mathrm{hp}}(\lambda)$ shape how the layer responds to each graph eigenvalue λ_j, exactly as in standard spectral GNNs [6, 29].

2. *Chaos-order coupling*: the matrices $M^{(\ell,\mathrm{lp})}$ and $M^{(\ell,\mathrm{hp})}$ control the coupling between chaos orders $0, \dots, P$, enabling the layer to redistribute energy across the chaos spectrum.

Together, these two axes parameterize a bilinear family of operators on the tensor-product Hilbert space $\ell^2(V) \otimes L^2(\mathbb{R}, \gamma)$, thereby justifying the *doubly-spectral* designation. □

A.5 Proof of Theorem 2 (Representational capacity and truncation)

Theorem 2 (Representational capacity and truncation under full-rank features)**.** *Let $\mathcal{G}$ have N nodes and features $X \in \mathbb{R}^{N\times d_{\mathrm{in}}}$ with $\mathrm{rank}(X) = N$ (which requires $d_{\mathrm{in}} \geq N$). With ReLU activation, a restricted subfamily of DSS-GNNs (one layer, lifted width $2d(P{+}1)$, quadrature size $S = P{+}1$, constant gates, identity low-pass filter) has the following properties. For any target $Y \in L^2(\mathbb{R}, \gamma; \mathbb{R}^{N\times d})$ and any chaos order P, some output $\widetilde{Y}_P$ matches the chaos coefficients of Y through order P; letting P and width vary, the resulting outputs are dense in $L^2(\mathbb{R}, \gamma; \mathbb{R}^{N\times d})$. If Y in addition admits an entire Gaussian extension with controlled growth (Appendix A.6), the truncation error decays exponentially:*

$$\|Y - \widetilde{Y}_P\|_{L^2(\mathbb{R},\gamma;\,\mathbb{R}^{N\times d})} \leq C_{\mathrm{tr}}\, \beta^P \tag{10}$$

for some $C_{\mathrm{tr}} > 0$ and $\beta \in (0,1)$ depending on Y.

Proof of the approximation claim. The proof proceeds in three stages: (i) truncation in the chaos-order axis, (ii) a rank argument for the quadrature projection, and (iii) a one-layer constructive realization using the final linear readout.

Step 1 (Chaos-order truncation). By the Wiener–Itô chaos decomposition [14], any $Y \in L^2(\mathbb{R}, \gamma; \mathbb{R}^{N\times d})$ admits the expansion $Y(\omega) = \sum_{n=0}^{\infty} Y_n \Psi_n(\omega)$ with deterministic coefficients $Y_n = \mathbb{E}[Y\Psi_n] \in \mathbb{R}^{N\times d}$, and by Parseval's identity $\|Y\|_{L^2}^2 = \sum_{n=0}^{\infty} \|Y_n\|_F^2 < \infty$. Fix a truncation order P for the construction. For the density claim, fix $\varepsilon > 0$ and take this P large enough that $\sum_{n>P} \|Y_n\|_F^2 < \varepsilon^2$; for the explicit rate, keep P arbitrary and apply the tail estimate proved below.

Step 2 (Quadrature projection rank). Take $S = P + 1$ and let $\{(\omega_s, \mu_s)\}_{s=1}^{S}$ be the associated S-point Gauss–Hermite rule. Define the projection matrix $\Phi \in \mathbb{R}^{(P+1)\times S}$ with entries $\Phi_{ns} = \mu_s\, \Psi_n(\omega_s)$. The Hermite polynomials $\Psi_0, \dots, \Psi_P$ are linearly independent on the $S = P + 1$

distinct quadrature nodes, and $\mu_s > 0$ for all s, so Φ is invertible. Consequently, for any target coefficients $\{Y_n\}_{n=0}^{P}$, there exist matrices $\{V_s\}_{s=1}^{S}$ in $\mathbb{R}^{N\times d}$ satisfying

$$\sum_{s=1}^{S} \mu_s \, V_s \, \Psi_n(\omega_s) = Y_n, \qquad n = 0, \dots, P. \tag{28}$$

Step 3 (Constructive one-layer realization with ReLU). We construct a one-layer DSS-GNN ($L = 1$) whose post-readout output matches the truncated coefficients exactly. Let $q := 2d$ and define the nonnegative hidden targets

$$R_s := [V_s^+, \, V_s^-] \in \mathbb{R}_+^{N\times q}, \qquad V_s = V_s^+ - V_s^-,$$

where V_s^+, V_s^- are the entrywise positive and negative parts of V_s. Choose the final readout

$$W_{\text{out}} = \begin{bmatrix} I_d \\ -I_d \end{bmatrix} \in \mathbb{R}^{q\times d},$$

so that $R_s W_{\text{out}} = V_s$ for every s.

Next define the generalized Vandermonde matrix

$$\Theta \in \mathbb{R}^{S\times S}, \qquad \Theta_{st} := \Psi_{t-1}(\omega_s), \qquad s, t = 1, \dots, S.$$

Because $\Psi_0, \dots, \Psi_{S-1}$ are linearly independent and the quadrature nodes $\omega_1, \dots, \omega_S$ are distinct, Θ is invertible. Hence for the prescribed matrices R_s there exist unique matrices $A_1, \dots, A_S \in \mathbb{R}^{N\times q}$ such that

$$\sum_{t=1}^{S} \Psi_{t-1}(\omega_s) \, A_t = R_s, \qquad s = 1, \dots, S. \tag{29}$$

Next, we realize these A_t with the DSS layer. Set the lifted input width to $d_0 = Sq$ and partition the lifted features into S blocks $b_1, \dots, b_S$, each of width q. Choose the graph filters and gates as

$$G_{\text{lp}}(L_G) = I \quad (K_{\text{lp}} = 0, \; c_0^{(\text{lp})} = 1), \qquad G_{\text{hp}}(L_G) = 0,$$
$$\alpha_n(\omega) \equiv 1 \;\; (n = 0, \dots, P), \qquad \beta_n(\omega) \equiv 0,$$

so $P_g = 0$ suffices. Since $\operatorname{rank}(X) = N$, for each $t = 1, \dots, S$ there exists a matrix $B_t \in \mathbb{R}^{d_{\text{in}}\times q}$ with $XB_t = A_t$. Set $W_{\text{in}}^{(t-1)}$ to place B_t in block b_t and zeros elsewhere. Then $H_{t-1}^{(0)} = XW_{\text{in}}^{(t-1)}$ is zero outside block b_t and equals A_t on block b_t.

Finally, choose $W_0 \in \mathbb{R}^{d_0\times q}$ so that each row-block corresponding to b_t equals the identity I_q. At quadrature node ω_s, the quadrature-node pre-activation becomes

$$\widetilde{H}^{(1)}(\omega_s) = \sum_{t=1}^{S} \Psi_{t-1}(\omega_s) \, A_t = R_s$$

by (29). Since each R_s is entrywise nonnegative and $\sigma = \text{ReLU}$, we obtain

$$\sigma\big(\widetilde{H}^{(1)}(\omega_s)\big) = R_s, \qquad \sigma\big(\widetilde{H}^{(1)}(\omega_s)\big) W_{\text{out}} = V_s.$$

Projecting back to chaos coefficients gives, for every $m = 0, \dots, P$,

$$H_m^{(1)} W_{\text{out}} = \sum_{s=1}^{S} \mu_s \, \sigma\big(\widetilde{H}^{(1)}(\omega_s)\big) W_{\text{out}} \Psi_m(\omega_s) = \sum_{s=1}^{S} \mu_s \, V_s \, \Psi_m(\omega_s) = Y_m,$$

where the last equality is exactly (28).

Step 4 (Combining with the tail). The retained coefficients are matched exactly, so the order-P output $\widetilde{Y}_P$ lies in $\bigoplus_{n=0}^{P} \mathcal{C}_n \otimes \mathbb{R}^{N\times d}$ and has no chaos coefficients for $n > P$. By orthogonality of the chaos subspaces and Parseval's identity, it satisfies

$$\|Y - \widetilde{Y}_P\|_{L^2}^2 = \sum_{n>P} \|Y_n\|_F^2.$$

Thus choosing P so that the right-hand side is below ε^2 proves density in L^2. □

Remark 4 (Practical vs. theoretical capacity)**.** *The constructive part of Theorem 2 is an existence result for a restricted subfamily of DSS-GNNs: the proof requires* $\operatorname{rank}(X) = N$ *(i.e.,* $d_{\text{in}} \geq N$*), a one-layer construction with lifted width* $d_0 = 2dS = 2d(P{+}1)$ *and output width* $d_1 = 2d$*, quadrature size* $S = P + 1$*, the ReLU activation, constant gates (*$P_g = 0$*), the identity low-pass filter, and a final linear readout that combines positive and negative hidden channels. The construction uses the identity low-pass filter and constant gates, so it bounds the capacity of the chaos axis alone; Theorem 1 describes the graph-frequency axis. The practical model differs from these conditions (*d_{in} *is typically smaller than* N*,* P *is small, and the filters are learned), so the theorem establishes representational capacity, while calibration, OOD detection, and robustness are established empirically. The explicit exponential bound adds that, for targets satisfying the entire-extension condition, a low truncation order should capture most of the gain, which is empirically consistent with the plateau at* $P = 1$–2 *observed in Figure 3(b).*

Remark 5 (Rank-deficient features)**.** *When* $\operatorname{rank}(X) = r < N$*, the same construction still matches every truncated target whose chaos-coefficient columns lie in* $\operatorname{range}(X)$*, under one positivity condition required by the ReLU step: the nonnegative quadrature-node evaluations* $R_s = [V_s^+, V_s^-]$ *of Step 3 must themselves have columns in* $\operatorname{range}(X)$*, since* $A_t = XB_t$ *forces every* A_t*, and hence every* R_s*, into that range. The condition is satisfiable, for example, by appending a constant feature column to* X*: with* $V_s^+ := V_s + c\,\mathbf{1}\mathbf{1}^\top$ *and* $V_s^- := c\,\mathbf{1}\mathbf{1}^\top$ *for* $c \geq \max_s \|V_s\|_{\max}$*, both parts are nonnegative and have columns in* $\operatorname{range}([X, \mathbf{1}]) \supseteq \operatorname{range}(X)$*, so every truncated target with columns in* $\operatorname{range}([X, \mathbf{1}])$ *is matched. We state exactly this matching property; no claim is made for targets whose coefficients are outside that range.*

A.6 Proof of the Truncation Bound in Theorem 2

The analytic Gaussian-growth condition referenced in Theorem 2 is the following: each centered coordinate $\bar{Y}_{ik}(\omega) := Y_{ik}(\omega) - \mathbb{E}[Y_{ik}]$ extends to an entire function satisfying $|\bar{Y}_{ik}(z)| \leq A_{ik} \exp(a|z|^2)$ for some $a \in (0, \frac{1}{4})$ uniform in (i, k), with $\sum_{i,k} A_{ik}^2 < \infty$. Under this condition, set $\alpha := 2a/(1-2a) < 1$; the proof yields the theorem for every $\beta \in (\sqrt{\alpha}, 1)$, with C_{tr} depending on a, β, and $\sum_{i,k} A_{ik}^2$.

Proof of the truncation bound. We prove the bound coordinate-wise and sum. Fix a centered coordinate function $g := \bar{Y}_{ik}$ with chaos coefficients $c_n := \mathbb{E}[g(\omega)\Psi_n(\omega)]$, so that $g(\omega) = \sum_{n=1}^{\infty} c_n \Psi_n(\omega)$.

Step 1 (Hermite generating function). The generating function of the normalized Hermite polynomials [32] gives $\sum_{n=0}^{\infty} \frac{t^n}{\sqrt{n!}} \Psi_n(\omega) = \exp(t\omega - t^2/2)$. Multiplying by $g(\omega)$ and taking the Gaussian expectation yields

$$\phi(s) \;:=\; \sum_{n=1}^{\infty} \frac{c_n}{\sqrt{n!}}\, s^n \;=\; \mathbb{E}\big[g(\omega + s)\big] \;=\; \int_{\mathbb{R}} g(x+s)\, d\gamma(x), \tag{30}$$

which is verified by expanding g in the Hermite basis and using $\mathbb{E}[\Psi_n(\omega)\exp(s\omega - s^2/2)] = s^n/\sqrt{n!}$. The series defining ϕ converges for every complex s because $|c_n| \leq \|g\|_{L^2(\gamma)}$, so ϕ is entire; the right-hand side is entire by the growth bound below and dominated convergence, and the two sides agree for real s, hence for all $s \in \mathbb{C}$.

Step 2 (Growth bound on ϕ). The hypothesis $|g(z)| \leq A\exp(a|z|^2)$ with $a < \frac{1}{4}$ gives

$$|\phi(s)| \leq \int_{\mathbb{R}} A\, \exp\big(a|x+s|^2\big)\, d\gamma(x) = A\, e^{a|s|^2} \int_{\mathbb{R}} \exp\big(ax^2 + 2a\, x\, \operatorname{Re}(s)\big)\, d\gamma(x).$$

Completing the square in the exponent ($ax^2 + 2a\operatorname{Re}(s)\, x - x^2/2 = -(\frac{1}{2} - a)x^2 + 2a\operatorname{Re}(s)\, x$, which is a Gaussian integral convergent for $a < \frac{1}{2}$) yields

$$|\phi(s)| \;\leq\; \frac{A}{\sqrt{1-2a}}\, \exp\Big(\frac{a}{1-2a}\, |s|^2\Big).$$

Step 3 (Cauchy estimate and Stirling bound). Since ϕ is entire, Cauchy's inequality on the circle $|s| = R$ gives

$$\frac{|c_n|}{\sqrt{n!}} \;\leq\; R^{-n} \sup_{|s|=R} |\phi(s)| \;\leq\; \frac{A}{\sqrt{1-2a}}\, R^{-n}\, e^{a' R^2}, \qquad a' := \frac{a}{1-2a}.$$

Setting $R = \sqrt{n/(2a')}$ (the minimizer of $-n \ln R + a' R^2$):

$$|c_n| \;\le\; \frac{A}{\sqrt{1-2a}}\,\sqrt{n!}\,\Big(\frac{2a'}{n}\Big)^{n/2} e^{n/2}.$$

By Stirling's inequality $\sqrt{n!} \le e^{1/2}(2\pi n)^{1/4}(n/e)^{n/2}$, the product $(n/e)^{n/2}(2a'/n)^{n/2}e^{n/2}$ simplifies to $(2a')^{n/2}$, giving

$$c_n^2 \;\le\; \frac{A^2\, e\sqrt{2\pi n}}{1-2a}\,(2a')^n \;=\; \frac{A^2\, e\sqrt{2\pi n}}{1-2a}\,\alpha^n, \qquad \alpha := 2a' = \frac{2a}{1-2a}. \tag{31}$$

Since $a < \frac{1}{4}$, we have $\alpha < 1$.

Step 4 (Tail bound). For any $\eta \in (\alpha, 1)$ the $\sqrt{n}$ factor in (31) is absorbed into the geometric ratio: $\sqrt{n}\,\alpha^n \le C_\eta\,\eta^n$, where $C_\eta := \sup_{n\ge 1}\sqrt{n}\,(\alpha/\eta)^n < \infty$ since $\alpha/\eta < 1$. Substituting this into (31) and summing the resulting geometric series over $n > P$:

$$\sum_{n=P+1}^{\infty} c_n^2 \;\le\; \frac{A^2\, e\sqrt{2\pi}\, C_\eta}{1-2a}\,\frac{\eta^{P+1}}{1-\eta}.$$

Summing over all coordinates (i,k) and applying Parseval on the graph eigenbasis ($\sum_j \|\hat{Y}_{n,j}\|^2 = \|Y_n\|^2_{\ell^2(V)} = \sum_{i,k} c_{n,ik}^2$) [29]:

$$\sum_{n=P+1}^{\infty}\sum_{j=1}^{N} \|\hat{Y}_{n,j}\|^2 \;\le\; C\,\eta^P$$

for a constant C depending on a, η, and $\sum_{i,k} A_{ik}^2$. Now fix any $\beta \in (\sqrt{\alpha}, 1)$ and set $\eta = \beta^2$. Taking square roots gives

$$\Big(\sum_{n>P}\sum_{j=1}^{N} \|\hat{Y}_{n,j}\|^2\Big)^{1/2} \le C_{\mathrm{tr}}\beta^P, \qquad C_{\mathrm{tr}} := \sqrt{C}.$$

Since the construction above matches all retained coefficients exactly, $\|Y - \widetilde{Y}_P\|_{L^2}$ equals this omitted tail, which gives the bound stated in Theorem 2. Since $\beta > \sqrt{\alpha}$ can be chosen arbitrarily close to $\sqrt{\alpha}$, the effective decay rate is e^{-cP} with $c = \frac{1}{2}\ln(1/\alpha) = \frac{1}{2}\ln\big(\frac{1-2a}{2a}\big) > 0$. □

A.7 Proof of Proposition 2 (Stochastic and mean-logit readouts)

Lemma 3 (Readout regularity and quadrature conditions)**.** *Let $z_i^{(s)} = \sum_{n=0}^{P} Z_{i,n}\Psi_n(\omega_s)$, where $Z_{i,0}$ is the mean logit. Assume $S \ge P+1$, so the quadrature rule integrates $\Psi_n\Psi_m$ exactly for $n, m \le P$, and assume $\ell(\cdot, y_i)$ is C^3 on a convex neighborhood of $Z_{i,0}$ containing all quadrature logits. Then, with $\delta_i^{(s)} := z_i^{(s)} - Z_{i,0}$,*

$$\sum_s \mu_s\,\delta_i^{(s)} = 0, \qquad \sum_s \mu_s\,\delta_i^{(s)}(\delta_i^{(s)})^\top = \Sigma_{z_i},$$

and Taylor expansion around $Z_{i,0}$ has a remainder of order $O(\max_s \|\delta_i^{(s)}\|^3)$. For cross-entropy, $\nabla_z^2\ell(z,y) = \mathrm{diag}(p) - pp^\top$ with $p = \mathrm{softmax}(z)$; for finite logits its nullspace is $\mathrm{span}\{\mathbf{1}\}$.

Proof. The first identity follows from $\sum_s \mu_s\Psi_n(\omega_s) = \mathbb{E}[\Psi_n] = 0$ for $n \ge 1$. The second follows from exact quadrature orthonormality, $\sum_s \mu_s\Psi_n(\omega_s)\Psi_m(\omega_s) = \delta_{nm}$, giving $\sum_s \mu_s\delta_i^{(s)}(\delta_i^{(s)})^\top = \sum_{n=1}^{P} Z_{i,n}Z_{i,n}^\top = \Sigma_{z_i}$. The C^3 condition gives the stated Taylor remainder. The cross-entropy Hessian is the covariance matrix of a categorical distribution with probabilities p, so its finite-logit nullspace consists exactly of common shifts. □

Proposition 2 (Stochastic and mean-logit readouts)**.** *Assume the number of Gauss–Hermite quadrature nodes is at least one more than the chaos order, and that the per-node loss $\ell(\cdot, y_i)$ satisfies the regularity conditions in Lemma 3. Then:*

1. *(Calibration channel.) The per-node quadrature-averaged loss* $\bar{\ell}_i := \sum_{s=1}^{S} \mu_s \, \ell(z_i^{(s)}, y_i)$ *admits the Taylor expansion* $\bar{\ell}_i = \ell(Z_{i,0}, y_i) + \frac{1}{2} \operatorname{tr}(\mathcal{H}_i \, \Sigma_{z_i}) + O\big(\max_s \|z_i^{(s)} - Z_{i,0}\|^3\big)$, *where* $\mathcal{H}_i = \nabla_z^2 \ell(Z_{i,0}, y_i)$ *is the loss Hessian and* Σ_{z_i} *is the chaos covariance* (7)*.*

2. *(OOD channel.) Shifting every class logit by the same constant leaves the softmax output, and hence any softmax-derived summary such as predictive entropy or mutual information, unchanged; by contrast, the energy score* (9) *decreases by exactly that constant.*

Proof. The quadrature logits at node ω_s are $z_i^{(s)} = \sum_{n=0}^{P} Z_{i,n} \Psi_n(\omega_s)$, and the quadrature-averaged loss is $\bar{\ell}_i = \sum_{s=1}^{S} \mu_s \, \ell(z_i^{(s)}, y_i)$.

Step 1 (Taylor expansion at each quadrature node). Write $z_i^{(s)} = Z_{i,0} + \delta_i^{(s)}$ where $\delta_i^{(s)} = \sum_{n=1}^{P} Z_{i,n} \Psi_n(\omega_s) \in \mathbb{R}^C$ is the stochastic perturbation. Expanding ℓ to second order around $Z_{i,0}$, and using that $\ell(\cdot, y_i)$ is C^3 in a neighborhood of $Z_{i,0}$:

$$\ell(z_i^{(s)}, y_i) = \ell(Z_{i,0}, y_i) + \nabla_z \ell(Z_{i,0}, y_i)^\top \delta_i^{(s)} + \tfrac{1}{2} (\delta_i^{(s)})^\top \nabla_z^2 \ell(Z_{i,0}, y_i) \, \delta_i^{(s)} + O(\|\delta_i^{(s)}\|^3).$$

Step 2 (Quadrature averaging of the linear term). The first-order term vanishes under quadrature averaging:

$$\sum_{s=1}^{S} \mu_s \, \nabla_z \ell^\top \delta_i^{(s)} = \nabla_z \ell^\top \sum_{n=1}^{P} Z_{i,n} \underbrace{\sum_{s=1}^{S} \mu_s \, \Psi_n(\omega_s)}_{= \mathbb{E}[\Psi_n] = 0} = 0,$$

since $\mathbb{E}[\Psi_n] = 0$ for $n \geq 1$ by the orthonormality of the Hermite basis (14), and the Gauss–Hermite rule integrates Ψ_n exactly for $n \leq 2S - 1$.

Step 3 (Quadrature averaging of the quadratic term). For the second-order term, abbreviate $\mathcal{H}_i := \nabla_z^2 \ell(Z_{i,0}, y_i)$ and compute:

$$\begin{aligned}
\sum_{s=1}^{S} \mu_s \, (\delta_i^{(s)})^\top \mathcal{H}_i \, \delta_i^{(s)} &= \sum_{s=1}^{S} \mu_s \Big(\sum_{n=1}^{P} Z_{i,n} \Psi_n(\omega_s) \Big)^\top \mathcal{H}_i \Big(\sum_{m=1}^{P} Z_{i,m} \Psi_m(\omega_s) \Big) \\
&= \sum_{n,m=1}^{P} Z_{i,n}^\top \mathcal{H}_i \, Z_{i,m} \underbrace{\sum_{s=1}^{S} \mu_s \, \Psi_n(\omega_s) \Psi_m(\omega_s)}_{= \delta_{nm}} \\
&= \sum_{n=1}^{P} Z_{i,n}^\top \mathcal{H}_i \, Z_{i,n} \;=\; \operatorname{tr}\Big(\mathcal{H}_i \sum_{n=1}^{P} Z_{i,n} Z_{i,n}^\top \Big) \;=\; \operatorname{tr}(\mathcal{H}_i \, \Sigma_{z_i}),
\end{aligned}$$

where the orthonormality $\sum_s \mu_s \Psi_n(\omega_s) \Psi_m(\omega_s) = \delta_{nm}$ holds by exactness of the Gauss–Hermite rule for polynomials of degree $n + m \leq 2P \leq 2S - 1$, and $\Sigma_{z_i} = \sum_{n=1}^{P} Z_{i,n} Z_{i,n}^\top$ is the logit covariance from (7).

Step 4 (Combining). Substituting Steps 2 and 3 into the averaged Taylor expansion:

$$\bar{\ell}_i = \sum_s \mu_s \, \ell(z_i^{(s)}, y_i) = \ell(Z_{i,0}, y_i) + 0 + \tfrac{1}{2} \operatorname{tr}\big(\nabla_z^2 \ell(Z_{i,0}, y_i) \, \Sigma_{z_i} \big) + O\Big(\max_s \|\delta_i^{(s)}\|^3 \Big). \tag{32}$$

Step 5 (Positive semi-definiteness and nullspace characterization for cross-entropy). For the cross-entropy loss $\ell(z, y) = -\log p_y$ where $p = \operatorname{softmax}(z)$, the Hessian is $\nabla_z^2 \ell = \operatorname{diag}(p) - pp^\top$. This is the covariance matrix of a categorical distribution with probabilities p, hence positive semi-definite: for any $v \in \mathbb{R}^C$, $v^\top (\operatorname{diag}(p) - pp^\top) v = \mathbb{E}_{\hat{y} \sim p}[v_{\hat{y}}^2] - (\mathbb{E}_{\hat{y} \sim p}[v_{\hat{y}}])^2 = \operatorname{Var}_{\hat{y} \sim p}(v_{\hat{y}}) \geq 0$. Since $\mathcal{H}_i \succeq 0$ and $\Sigma_{z_i} \succeq 0$,

$$\operatorname{tr}(\mathcal{H}_i \Sigma_{z_i}) = \operatorname{tr}\big(\Sigma_{z_i}^{1/2} \mathcal{H}_i \Sigma_{z_i}^{1/2} \big) \geq 0.$$

Moreover, $\operatorname{tr}(\mathcal{H}_i \Sigma_{z_i}) = 0$ if and only if $\Sigma_{z_i}^{1/2} \mathcal{H}_i \Sigma_{z_i}^{1/2} = 0$, equivalently $\operatorname{range}(\Sigma_{z_i}) \subseteq \operatorname{null}(\mathcal{H}_i)$. For finite logits, softmax probabilities satisfy $p_c > 0$ for every class, and the Hessian nullspace is

$\mathrm{null}(\mathcal{H}_i) = \mathrm{span}\{\mathbf{1}\}$; thus purely common logit offsets do not contribute to the correction. The special case $\Sigma_{z_i} = 0$ recovers the non-stochastic model, while in the fully confident limit $p \to e_y$ we have $\mathcal{H}_i \to 0$.

Step 6 (Energy shift identity). For any scalar a,

$$\mathrm{softmax}(z + a\mathbf{1})_c = \frac{e^{z_c+a}}{\sum_{r=1}^{C} e^{z_r+a}} = \frac{e^{z_c}}{\sum_{r=1}^{C} e^{z_r}} = \mathrm{softmax}(z)_c.$$

Thus any uncertainty summary that is a function only of the softmax probabilities, such as predictive entropy, is invariant under common-logit shifts. By contrast,

$$s_{\mathrm{energy}}(z + a\mathbf{1}) = -\log \sum_{c=1}^{C} e^{z_c+a} = -\log\Big(e^a \sum_{c=1}^{C} e^{z_c}\Big) = s_{\mathrm{energy}}(z) - a. \tag{33}$$

For completeness, the analogous second-order expansion for the quadrature-averaged energy follows by the same calculation. Write $z_i^{(s)} = Z_{i,0} + \delta_i^{(s)}$ and $A(z) = \log \sum_c e^{z_c}$. The gradient and Hessian of A are

$$\nabla A(z) = p, \qquad \nabla^2 A(z) = \mathrm{diag}(p) - pp^\top =: H_A(z),$$

where $p = \mathrm{softmax}(z)$. Since $s_{\mathrm{energy}} = -A$, Taylor expansion around $Z_{i,0}$ gives

$$s_{\mathrm{energy}}(z_i^{(s)}) = s_{\mathrm{energy}}(Z_{i,0}) - p_i^\top \delta_i^{(s)} - \tfrac{1}{2}(\delta_i^{(s)})^\top H_A(Z_{i,0})\delta_i^{(s)} + O(\|\delta_i^{(s)}\|^3).$$

The quadrature average of the linear term vanishes as in Step 2, and the quadratic term averages as in Step 3:

$$\sum_s \mu_s (\delta_i^{(s)})^\top H_A(Z_{i,0})\delta_i^{(s)} = \mathrm{tr}\big(H_A(Z_{i,0})\Sigma_{z_i}\big),$$

using exact quadrature orthonormality for pairwise products $\Psi_n \Psi_m$ with $n, m \le P$. Therefore

$$\sum_s \mu_s\, s_{\mathrm{energy}}(z_i^{(s)}) = s_{\mathrm{energy}}(Z_{i,0}) - \tfrac{1}{2}\,\mathrm{tr}\big(H_A(Z_{i,0})\,\Sigma_{z_i}\big) + O\Big(\max_s \|\delta_i^{(s)}\|^3\Big). \tag{34}$$

The Hessian $H_A(z)$ is the same categorical covariance matrix as the cross-entropy Hessian, so the stochastic energy correction also vanishes for common-logit stochastic variation. The non-stochastic common shift of the mean logit, however, is read out linearly by (33). □

Corollary 1 (Readout gap)**.** *Assume $S \ge P + 1$ as in Lemma 3. Let $f(z) = \mathrm{softmax}(z)$ and $B := \sup_{z\in\mathbb{R}^C} \max_c \|\nabla_z^2 f_c(z)\|_{\mathrm{op}}$, which is finite because every entry of $\nabla_z^2 f_c$ is a polynomial in the probabilities $f(z) \in [0,1]^C$; since $\nabla_z^2 f_c = f_c\big[(e_c - f)(e_c - f)^\top - (\mathrm{diag}(f) - ff^\top)\big]$ is f_c times a difference of positive semidefinite matrices of operator norm at most $2(1 - f_c)^2$ and $\frac{1}{2}$, $B \le \frac{1}{2}$ for every C. Then for every node i,*

$$\|\bar{p}_i - \mathrm{softmax}(Z_{i,0})\|_\infty \;\le\; \tfrac{B}{2}\,\mathrm{tr}(\Sigma_{z_i}),$$

and $\arg\max_c \bar{p}_{i,c} = \arg\max_c \mathrm{softmax}(Z_{i,0})_c$ whenever the gap between the two largest entries of $\mathrm{softmax}(Z_{i,0})$ exceeds $B\,\mathrm{tr}(\Sigma_{z_i})$.

Proof. Write $z_i^{(s)} = Z_{i,0} + \delta_i^{(s)}$. Taylor's theorem with integral remainder gives, for each class c,

$$f_c(z_i^{(s)}) = f_c(Z_{i,0}) + \nabla f_c(Z_{i,0})^\top \delta_i^{(s)} + \int_0^1 (1-t)\,(\delta_i^{(s)})^\top \nabla^2 f_c\big(Z_{i,0} + t\delta_i^{(s)}\big)\,\delta_i^{(s)}\,dt,$$

and the remainder is bounded in absolute value by $\frac{B}{2}\|\delta_i^{(s)}\|^2$. Averaging with the quadrature weights, the linear term vanishes because $\sum_s \mu_s \delta_i^{(s)} = 0$ (Lemma 3), and $\sum_s \mu_s \|\delta_i^{(s)}\|^2 = \mathrm{tr}(\Sigma_{z_i})$ by the same lemma, which gives the bound. Each component of $\bar{p}_i$ therefore lies within $\frac{B}{2}\,\mathrm{tr}(\Sigma_{z_i})$ of the corresponding component of $\mathrm{softmax}(Z_{i,0})$, so the two argmax rules agree whenever the top-two gap of $\mathrm{softmax}(Z_{i,0})$ exceeds $B\,\mathrm{tr}(\Sigma_{z_i})$. Since (11) penalizes $\mathcal{E}_i = \mathrm{tr}(\Sigma_{z_i})$ on training nodes, the bound is the structural reason the two readouts agree closely in practice (Section 4.2, Table 27). □

B Baseline Details

B.1 Baseline methods

Table 6 summarizes which tasks each method evaluates. G-ΔUQ is the closest baseline in scope because it reports all three task families; we compare against it on calibration (Table 1) and distribution shift (Table 4). On OOD detection, the anchor-sampling mechanism in G-ΔUQ causes out-of-memory errors in the GNNSafe OOD pipeline, preventing a direct comparison. Across the evaluated benchmarks, the DSS representation gives the strongest calibration and GOOD results among the compared methods, with the best AUROC on 7 of 11 OOD settings. The calibration and GOOD results use plain cross-entropy or ERM training without task-specific robust objectives; in the OOD detection experiments DSS-Hybrid is trained with the same energy-margin objective as GNNSafe++ (Appendix D).

Table 6: Tasks evaluated by each method. ✓= evaluated in the original paper; (✓) = evaluated as a secondary metric or in the appendix; ✓* = evaluated in the original paper but OOM on our benchmark; ✗= not evaluated in the original paper or not designed for that task.

Method	Calibration	OOD Detection	Robust Classif.
MC-dropout [10]	(✓)	✗	✗
Deep Ensembles [17]	✓	(✓)	✗
G-ΔUQ [33]	✓	✓*	✓
TFE-GNN [8]	(✓)	✗	✗
GNNSafe [36]	✗	✓	✗
GNNSafe++ [36]	✗	✓	✗
Graph-EBM [9]	(✓)	✓	✗
GPN [30]	✓	✓	✗
IRM [1]	✗	✗	✓
GroupDRO [27]	✗	✗	✓
TAR [38]	✗	✗	✓
DSS-GNN / DSS-Hybrid (ours)	✓	✓	✓

We compare against the following baselines across the three evaluation tasks.

Calibration baselines.

- **TFE-GNN** [8]: A spectral GNN that combines three complementary polynomial filters (low-pass, high-pass, and band-pass) via learned attention weights. It uses a propagate-first architecture where spectral filtering is applied once before an MLP backbone.
- **G-ΔUQ** [33]: An intrinsic uncertainty method that estimates predictive uncertainty through anchor distances in the learned embedding space, without requiring ensembles or Monte Carlo sampling.

OOD detection baselines.

- **GNNSafe** [36]: Uses the energy score $E(x) = -T \log \sum_c \exp(z_c/T)$ as an OOD detection signal, propagated over the graph adjacency to exploit neighborhood consistency. Does not require OOD exposure during training.
- **GNNSafe++** [36]: Extends GNNSafe with an energy-based regularization loss that uses OOD training data to separate in-distribution and OOD energy scores. Uses per-dataset margin hyperparameters $(m_{\text{in}}, m_{\text{out}}, \lambda)$.
- **Graph-EBM** [9]: Models the joint distribution of node features and labels using an energy-based model with graph-aware components, including local and group energy terms propagated through graph diffusion.
- **MC-dropout** [10] and **Deep Ensembles** [17]: Stochastic baselines on the same GCN backbone and training protocol, with $M{=}20$ dropout passes at test time and $M{=}5$ independently trained models respectively; both are scored with the same energy score and graph propagation as the energy-based methods (Appendix D).

- **GPN** [30]: Graph Posterior Network, an evidential model that propagates per-node Dirichlet evidence over the graph; we report the values obtained on this benchmark by Wu et al. [36].

Distribution-shift baselines. Table 4 evaluates the GOOD/TAR baselines under concept shift and adds G-ΔUQ. Results for ERM through TAR are taken from the TAR paper [38]. Representative baselines include:

- **ERM**: Standard empirical risk minimization without any robustness modification.
- **IRM** [1]: Invariant Risk Minimization, which penalizes features that are not invariant across training environments.
- **GroupDRO** [27]: Distributionally robust optimization that upweights the worst-performing group during training.
- **VREx** [16]: Variance Risk Extrapolation, which penalizes variance of risks across training environments.
- **TAR** [38]: Topology-Aware Dynamic Reweighting, which uses graph-local transport to reweight training nodes based on loss and neighborhood structure.

C Dataset Statistics and Licenses

Tables 8–10 summarize the datasets used in our experiments. Table 7 lists the source repositories and licenses for all datasets. All datasets are publicly available and were used in accordance with their respective licenses.

Table 7: Dataset sources and licenses.

Dataset(s)	Repository	License
Cora, Citeseer, PubMed	`https://github.com/kimiyoung/planetoid`	MIT
Texas, Cornell, Wisconsin, Chameleon, Squirrel	`https://github.com/bingzhewei/geom-gcn`	not stated
Coauthor-CS, Amazon-Photo	`https://github.com/shchur/gnn-benchmark`	MIT
Roman-Empire, Amazon-Ratings, Minesweeper, Tolokers, Questions	`https://github.com/yandex-research/heterophilous-graphs`	MIT
Twitch	`https://github.com/benedekrozemberczki/MUSAE`	GPL-3.0
ogbn-arxiv	`https://github.com/snap-stanford/ogb`	MIT
GOOD benchmark	`https://github.com/divelab/GOOD`	GPL-3.0

C.1 Calibration benchmarks

Table 8: Node classification datasets for calibration experiments (Section 4.2). Results use 10 splits per dataset (Appendix D); edges are counted in both directions. Cornell uses the corrected files of the geom-gcn repository, whose initial release duplicated the Texas features and labels. Datasets below the mid-rule are from the heterophilous suite of Platonov et al. [26].

Dataset	Nodes	Edges	Classes	Features
Cora	2,708	10,556	7	1,433
Citeseer	3,327	9,228	6	3,703
PubMed	19,717	88,651	3	500
Texas	183	574	5	1,703
Cornell	183	557	5	1,703
Wisconsin	251	916	5	1,703
Chameleon	2,277	62,792	5	2,325
Squirrel	5,201	396,846	5	2,089
CS	18,333	163,788	15	6,805
Roman-Empire	22,662	65,854	18	300
Amazon-Ratings	24,492	186,100	5	300
Minesweeper	10,000	78,804	2	7
Tolokers	11,758	1,038,000	2	10
Questions	48,921	307,080	2	301

C.2 OOD detection benchmarks

Table 9: OOD detection datasets (Section 4.3). For node-OOD (Cora, Amazon-Photo, Coauthor-CS), OOD nodes are generated via structure (stochastic block model), feature (random interpolation of node features), or label (held-out classes) perturbations. For cross-graph OOD, the ID and OOD graphs are from different domains. Statistics are for the in-distribution graph.

Dataset	OOD type	Nodes	Edges	Classes	Features
Cora	structure / feature / label	2,708	10,556	7	1,433
Amazon-Photo	structure / feature / label	7,650	238,162	8	745
Coauthor-CS	structure / feature / label	18,333	163,788	15	6,805
Twitch (DE→ES/FR/RU)	cross-graph	9,498	306,276	2	3,170
Arxiv ($\leq$2015→2018–20)	cross-graph (temporal)	169,343	2,315,598	40	128

C.3 GOOD distribution-shift benchmarks

Table 10 reports the node-level concept-shift settings from GOOD [12] used in our distribution-shift experiments.

Table 10: GOOD benchmark settings (Section 4.4). All use concept shift. The domain column indicates the covariate that defines the distribution shift. Node counts are for the full graph before splitting; GOOD-Cora is built on the full Cora citation graph (CoraFull), not on the 7-class Planetoid subset of Table 8.

Setting	Domain	Nodes	Edges	Classes	Features
GOOD-Cora / degree	node degree	19,793	126,842	70	8,710
GOOD-Cora / word	word diversity	19,793	126,842	70	8,710
GOOD-Arxiv / degree	node degree	169,343	2,315,598	40	128
GOOD-Arxiv / time	publication year	169,343	2,315,598	40	128
GOOD-CBAS / color	node color	700	3,962	4	4
GOOD-WebKB / university	university	617	1,138	5	1,703
GOOD-Twitch / language	user language	34,120	892,346	2	128

D Implementation Details

D.1 General training protocol

All DSS-GNN models are trained with the mean-logit cross-entropy loss $\mathcal{L}_{\text{base}}$ (11); the quadrature-averaged loss

$$\mathcal{L}_{\text{quad}} = \frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{i \in \mathcal{V}_{\text{tr}}} \bar{\ell}_i \; + \; \lambda_{\text{reg}} \frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{i \in \mathcal{V}_{\text{tr}}} \mathcal{E}_i, \qquad \bar{\ell}_i = \sum_{s=1}^{S} \mu_s \, \ell(z_i^{(s)}, y_i), \tag{35}$$

supervises every quadrature node, agrees with $\mathcal{L}_{\text{base}}$ to first order by Proposition 2, and is used only in that analysis. The loss includes the chaos energy regularizer; λ_{reg} is selected per dataset on validation Brier from $\{0, 10^{-3}, 10^{-2}, 10^{-1}\}$ (Table 12; Table 17 reports its effect at $P{=}2$), with 0.01 as the default. Training uses early stopping on validation loss with the patience listed per task below. All calibration results report mean $\pm$ standard deviation over 10 splits, identical for every method. For the nine datasets above the mid-rule of Table 8 the splits are random with fixed seeds: a class-balanced training set whose per-class quota totals 60% of the nodes when every class is large enough, 20% of the nodes for validation, and the remainder for testing. The five heterophilous-suite datasets use the first 10 public 50/25/25 splits of Platonov et al. [26].

D.2 Calibration (Table 1)

Shared defaults. Chebyshev filter degrees $K_{\text{lp}}{=}K_{\text{hp}}{=}4$, chaos order centered at $P{=}2$ in the base grid, quadrature nodes $S{=}4$, $\lambda_{\text{reg}}{=}0.01$, deterministic scalar gates ($P_g{=}0$), loss type = mean,

activation = ReLU, learning rate 0.01, weight decay 5×10^{-4}, propagation dropout 0.5, linear dropout 0.0, epochs 1000, patience 200. The best architecture variant per dataset was selected from a grid over propagation type (Chebyshev / TFE-adjacency / propagate-first), graph filter symmetry (symmetric / random-walk), optimizer (Adam / RMSprop), hidden dimension (64 / 128), number of layers (2 / 3), and dropout (0.5 / 0.6 / 0.8), evaluated on validation Brier score. Roman-Empire uses hidden 32 and $K_{\mathrm{lp}}{=}K_{\mathrm{hp}}{=}2$, outside this grid, Minesweeper uses learning rate 0.005, and Chameleon uses sum aggregation. The chaos order P is a regular hyperparameter selected per dataset on validation Brier over the sweep of Appendix E; Table 12 lists the selected order per dataset together with that single configuration's accuracy and Brier.

Per-dataset configurations. Table 11 lists the architecture variant and any hyperparameter overrides from the shared defaults for each dataset.

Table 11: Per-dataset non-P DSS-GNN configuration for the calibration experiments (Table 1). "Prop-first" = propagate-once structure (spectral filtering followed by MLP); "TFE-adj" = TFE-style adjacency propagation; "Cheb" = Chebyshev filters on rescaled Laplacian. Unlisted hyperparameters use the shared defaults above; P, λ_{reg}, and S per dataset are listed in Table 12.

Dataset	Architecture	Hidden	L	Optimizer	Overrides
Cora	Prop-first (rw)	64	2	Adam	dropout 0.8
Citeseer	Cheb	64	2	Adam	none
PubMed	Cheb	128	2	Adam	none
Texas	Prop-first	64	2	RMSprop	none
Cornell	Cheb	128	2	Adam	dropout 0.8
Wisconsin	Cheb	64	2	Adam	dropout 0.6
Chameleon	Prop-first (sum)	64	2	Adam	none
Squirrel	Prop-first (rw)	64	2	RMSprop	none
CS	TFE-adj	64	2	Adam	none
Roman-Emp.	TFE-adj	32	3	Adam	hidden 32, $K{=}2$
Amz-Rat.	TFE-adj	64	3	Adam	none
Minesweeper	Cheb	64	2	Adam	lr 0.005
Tolokers	Prop-first (rw)	64	2	RMSprop	none
Questions	Prop-first	64	2	RMSprop	none

Selected chaos order per dataset. Table 12 lists, for each dataset, the chaos order and regularization strength selected on validation Brier, with the accuracy and Brier of that configuration over the same 10 splits, taken from the sweeps of Tables 13 and 14, and its validation Brier (recorded in a separate run of the same configuration on the same splits); Table 1 reports these configurations. DSS-GNN has the best mean Brier on all 14 datasets, including the closest comparisons (Texas 0.240 vs 0.255, Squirrel 0.394 vs 0.395, Amazon-Ratings 0.634 vs 0.637, all TFE-GNN; Questions 0.053 vs 0.054, G-ΔUQ). At least one of $P{=}1$ or $P{=}2$ is within 0.02 Brier of the best swept order on every dataset (Section 4.2), so the larger selected orders are near-ties rather than requirements. The quadrature size is $S{=}4$ throughout: exact for the polynomial filter and gate terms at $P \leq 3$ (Appendix A.1), while at $P \geq 4$ the highest-order Hermite products are integrated approximately. The recommended default $P{=}1$–2 is exact for these terms at $S{=}4$, and the S sweep at $P{=}2$ (Table 18) moves Brier by at most 0.011 on 12 of 14 datasets.

D.3 OOD detection (Section 4.3)

DSS-Hybrid uses a 2-layer GCN base encoder (hidden 64, dropout 0.0) with a DSS residual branch ($K_{\mathrm{lp}}{=}3$, $K_{\mathrm{hp}}{=}2$, $P{=}1$, $S{=}4$, deterministic scalar gates). The base encoder is warmed up for 50 epochs before the DSS branch is activated; the residual scale γ_{res} is initialized to 0.1. All methods run in our pipeline use Adam with learning rate 0.01, weight decay 0.01, 200 epochs, and 3 seeds, following the GNNSafe protocol. The GNNSafe pipeline uses BatchNorm in its GCN backbones, and so does the DSS-Hybrid base; the standalone model of Appendix G adds per-chaos-channel BatchNorm in all OOD and GOOD cells but OOD Twitch and GOOD-Arxiv/degree. OOD detection scores are computed with the energy score (9) at temperature $T{=}1$, propagated over the graph adjacency with $K{=}2$ hops and mixing $\alpha{=}0.5$, following the GNNSafe protocol [36]. DSS-Hybrid is trained with the GNNSafe++ energy-margin regularizer, using the per-dataset margin hyperparameters

Table 12: Selected configuration per dataset for the calibration experiments (Table 1): chaos order P, chaos energy regularization λ_{reg}, quadrature size S, and that single configuration's test accuracy (%) and Brier score (mean $\pm$ SD over the same 10 splits). The last column reports the validation Brier of the selected configuration, recorded in a separate run on the same 10 splits (for Cornell, in the same run).

Dataset	P	λ_{reg}	S	Accuracy	Brier	Val. Brier
Cora	1	0.01	4	86.63 ± 1.26	0.207 ± 0.019	0.203
Citeseer	2	0	4	80.20 ± 1.28	0.308 ± 0.010	0.315
PubMed	5	0.01	4	89.72 ± 0.31	0.156 ± 0.005	0.152
Texas	2	0.01	4	91.31 ± 3.44	0.240 ± 0.111	0.190
Cornell	1	0.01	4	85.11 ± 5.12	0.231 ± 0.079	0.166
Wisconsin	2	0	4	93.25 ± 3.39	0.104 ± 0.047	0.088
Chameleon	1	0.01	4	75.36 ± 1.19	0.339 ± 0.019	0.333
Squirrel	1	0.1	4	68.06 ± 2.21	0.394 ± 0.027	0.415
CS	1	0.01	4	96.17 ± 0.23	0.062 ± 0.004	0.060
Roman-Empire	1	0	4	78.84 ± 0.61	0.300 ± 0.007	0.304
Amazon-Ratings	8	0.01	4	49.72 ± 0.64	0.634 ± 0.005	0.637
Minesweeper	6	0.001	4	87.66 ± 1.10	0.169 ± 0.014	0.170
Tolokers	2	0.01	4	79.88 ± 0.49	0.268 ± 0.005	0.305
Questions	4	0.001	4	97.17 ± 0.04	0.053 ± 0.001	0.054

$(m_{\text{in}}, m_{\text{out}}, \lambda)$ of Wu et al. [36], as GNNSafe++ is; Graph-EBM, MC-dropout, and Deep Ensembles are trained without it, as GNNSafe is. MC-dropout keeps dropout active at test time with BatchNorm frozen and aggregates $M{=}20$ stochastic passes; Deep Ensembles aggregate $M{=}5$ independently initialized backbones; both are scored with the same energy score and propagation as the energy-based methods, applied to the mean logits across passes or members.

Standalone model in the OOD and GOOD pipelines. For the cross-evaluation of Appendix G, the standalone model adds one BatchNorm1d per chaos channel between hidden DSS layers where noted; Table 1 uses no BatchNorm. The per-cell configurations are listed in Appendix G.

Graph-EBM reproduction. Graph-EBM is run with a port of the official `GraphEBMWrapper` [9] at the official default hyperparameters: the multi-scale corrected-conditional energy model with Mahalanobis logit correction, three energy scales, and label-propagation diffusion (diagonal untied covariance, $\gamma{=}1$, unit energy weights, $\alpha{=}0.5$ with $k{=}10$ steps). The scorer is fitted on the same GCN encoder (with BatchNorm, as in the GNNSafe pipeline), trained with the same hyperparameters and seeds as the other baselines run in our pipeline; Graph-EBM's own diffusion replaces the GNNSafe score propagation for this row. On an exported Cora/structure run (trained GCN logits and embeddings, both graphs, labels, and masks) the port and the official implementation agree at Spearman rank correlation 0.994, and exactly (rank correlation 1.0) once the interval and covariance estimators of the official implementation are used.

D.4 Distribution shift (Table 4)

DSS-Hybrid uses a 2-layer GCN base encoder with a DSS residual branch ($K_{\text{lp}}{=}3$, $K_{\text{hp}}{=}2$, $P{=}1$, $S{=}4$, hidden 64, deterministic scalar gates, 3 layers total). Training uses Adam with learning rate 10^{-3}, weight decay $5{\times}10^{-4}$, dropout 0.5, 500 epochs, patience 200. The base encoder is warmed up for 25 epochs. Standard ERM training is used (no robust objective). Published baseline results (ERM through TAR) are taken from the TAR paper [38]; G-ΔUQ is run in our pipeline.

D.5 Software, hardware, and compute budget

Experiments use PyTorch and PyTorch Geometric (the released environment pins PyTorch 2.5) on NVIDIA A100 (40 GB) and H100 GPUs; the ogbn-arxiv timing runs of Table 20 use one H100, and the Cornell experiments run on CPU. Mixed precision (FP16) is used for calibration experiments to reduce memory on larger graphs. On the A100 runs, per-epoch wall-clock times range from $\sim$35 ms (Cora, $P{=}0$) to $\sim$650 ms (Amazon-Ratings, $P{=}3$); see Table 20 for a detailed breakdown.

Compute estimate. The reported experiments (calibration: 14 datasets $\times$ 10 runs; OOD: 5 datasets $\times$ 3 seeds; GOOD: 7 settings) total approximately 8 A100 GPU-hours. The ablation studies (chaos order P, regularization λ_{reg}, quadrature nodes S, and contribution decomposition; Appendix E) add approximately 100 GPU-hours. Hyperparameter selection (grid search over architecture variant, hidden dimension, optimizer, and dropout) required an additional $\sim$25 GPU-hours. The total compute for the project, including preliminary experiments not reported in the paper, is approximately 200 A100 GPU-hours.

Code is available at `https://github.com/heraclixus/DSSGNN`.

E Ablation Study Details

E.1 Polynomial order P across all datasets

Figure 3 summarizes the chaos-order sweep referenced in Section 4.2, and Tables 13 and 14 report the per-dataset classification accuracy and Brier score for DSS-GNN across all 14 datasets for $P \in \{0, 1, 2, 3, 4, 5, 6, 8\}$. All results are mean $\pm$ standard deviation over 10 random splits. For each dataset, the best-performing P is highlighted in bold.

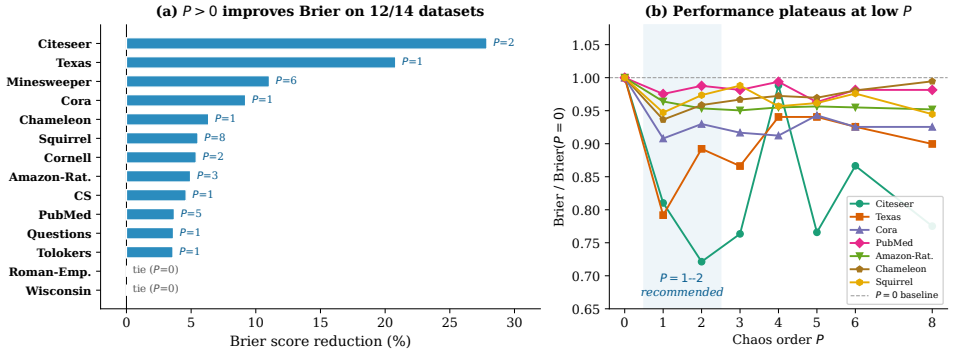

Figure 3: **(a)** Percentage Brier reduction of the best swept order $P{>}0$ over $P{=}0$ (Table 14). **(b)** Brier relative to $P{=}0$ versus chaos order, for seven datasets.

Table 13: Accuracy (%) of DSS-GNN for varying chaos order P across all datasets.

Dataset	$P=0$	$P=1$	$P=2$	$P=3$	$P=4$	$P=5$	$P=6$	$P=8$
Cora	85.12 ± 1.06	**86.63** ± 1.26	86.33 ± 1.12	86.33 ± 0.82	**86.63** ± 1.14	86.32 ± 1.13	86.61 ± 0.84	86.38 ± 0.96
Citeseer	65.39 ± 1.02	79.21 ± 0.79	**80.20** ± 1.28	78.47 ± 0.80	71.50 ± 4.58	78.92 ± 1.13	76.77 ± 2.19	78.49 ± 1.08
PubMed	89.12 ± 0.57	89.33 ± 0.49	89.33 ± 0.66	89.38 ± 0.67	89.33 ± 0.54	**89.72** ± 0.31	89.40 ± 0.57	89.39 ± 0.51
Texas	75.08 ± 23.70	88.36 ± 4.57	**91.31** ± 3.44	89.18 ± 4.92	**91.31** ± 3.30	87.54 ± 5.59	89.01 ± 2.82	87.37 ± 5.43
Cornell	83.62 ± 6.80	85.11 ± 5.12	**85.74** ± 6.73	83.62 ± 5.39	84.26 ± 6.87	83.83 ± 7.38	83.62 ± 6.53	82.98 ± 7.61
Wisconsin	**93.25** ± 3.46	92.87 ± 2.92	**93.25** ± 3.39	88.50 ± 3.35	93.00 ± 3.46	90.75 ± 2.84	89.63 ± 4.82	90.50 ± 2.92
Chameleon	74.53 ± 2.32	**75.36** ± 1.19	74.53 ± 1.74	74.51 ± 1.57	74.77 ± 1.86	74.92 ± 1.60	74.40 ± 1.65	73.63 ± 1.69
Squirrel	66.35 ± 2.56	**68.06** ± 2.21	67.26 ± 2.13	66.99 ± 1.24	67.39 ± 2.44	67.23 ± 1.55	67.02 ± 1.50	67.71 ± 1.51
CS	95.94 ± 0.20	**96.17** ± 0.23	96.01 ± 0.25	96.03 ± 0.19	96.13 ± 0.25	95.96 ± 0.29	96.02 ± 0.23	96.02 ± 0.17
Roman-Emp.	**79.01** ± 0.27	78.84 ± 0.61	78.81 ± 0.61	78.98 ± 0.87	78.75 ± 0.32	78.77 ± 0.69	**79.01** ± 0.27	78.97 ± 0.79
Amazon-Rat.	45.40 ± 0.76	48.13 ± 1.14	49.41 ± 0.71	49.65 ± 1.15	49.36 ± 0.70	49.05 ± 0.73	49.35 ± 0.83	**49.72** ± 0.64
Minesweeper	86.14 ± 0.60	86.93 ± 0.51	87.37 ± 1.10	86.37 ± 1.06	86.53 ± 0.96	86.17 ± 0.45	**87.66** ± 1.10	87.42 ± 0.93
Tolokers	79.33 ± 0.66	79.82 ± 0.45	**79.88** ± 0.49	79.28 ± 0.79	79.62 ± 0.56	79.32 ± 0.66	79.21 ± 0.75	79.36 ± 0.79
Questions	97.15 ± 0.06	97.15 ± 0.06	97.13 ± 0.06	97.16 ± 0.06	**97.17** ± 0.04	97.15 ± 0.06	97.15 ± 0.06	97.14 ± 0.09

On four datasets the best swept order exceeds two (PubMed, Squirrel, Amazon-Ratings, and Minesweeper); on every dataset $P = 1$ or $P = 2$ is within 0.02 Brier of the best swept value.

E.2 Chaos order on OOD detection and under distribution shift

OOD detection. Varying only the chaos order $P \in \{0, 1, 2, 3\}$ in the DSS-Hybrid OOD configuration (GCN base, energy score with propagation, energy-margin training, every other hyperparameter fixed; 3 runs) leaves energy AUROC flat: Cora-structure $91.0/91.2/90.9/90.8$ and Amazon-Photo/feature $99.4/99.5/99.5/99.5$ for $P = 0, 1, 2, 3$, Coauthor-CS likewise flat, and on ogbn-arxiv the AUROC moves by at most 0.8 across $P = 0$ to 3 (the runs of Table 20). The absolute

Table 14: Brier Score (↓) of DSS-GNN for varying chaos order P across all datasets.

Dataset	$P=0$	$P=1$	$P=2$	$P=3$	$P=4$	$P=5$	$P=6$	$P=8$
Cora	0.228 ± 0.013	**0.207** ± 0.019	0.212 ± 0.016	0.209 ± 0.016	0.208 ± 0.019	0.215 ± 0.015	0.211 ± 0.011	0.211 ± 0.015
Citeseer	0.427 ± 0.008	0.346 ± 0.008	**0.308** ± 0.010	0.326 ± 0.012	0.422 ± 0.027	0.327 ± 0.012	0.370 ± 0.023	0.331 ± 0.013
PubMed	0.162 ± 0.007	0.158 ± 0.007	0.160 ± 0.008	0.159 ± 0.008	0.161 ± 0.007	**0.156** ± 0.005	0.159 ± 0.008	0.159 ± 0.007
Texas	0.269 ± 0.082	**0.213** ± 0.111	0.240 ± 0.111	0.233 ± 0.096	0.253 ± 0.096	0.253 ± 0.086	0.249 ± 0.114	0.242 ± 0.082
Cornell	0.241 ± 0.087	0.231 ± 0.079	**0.228** ± 0.088	0.250 ± 0.090	0.263 ± 0.129	0.247 ± 0.107	0.252 ± 0.079	0.273 ± 0.111
Wisconsin	**0.103** ± 0.048	0.107 ± 0.041	0.104 ± 0.047	0.161 ± 0.060	**0.103** ± 0.048	0.129 ± 0.037	0.153 ± 0.052	0.135 ± 0.040
Chameleon	0.362 ± 0.019	**0.339** ± 0.019	0.347 ± 0.022	0.350 ± 0.016	0.352 ± 0.016	0.351 ± 0.018	0.355 ± 0.013	0.360 ± 0.015
Squirrel	0.416 ± 0.028	0.394 ± 0.027	0.405 ± 0.024	0.411 ± 0.019	0.398 ± 0.028	0.400 ± 0.016	0.406 ± 0.014	**0.393** ± 0.017
CS	0.065 ± 0.003	**0.062** ± 0.004	0.063 ± 0.003	0.064 ± 0.003	0.063 ± 0.003	0.064 ± 0.004	0.063 ± 0.003	0.063 ± 0.003
Roman-Emp.	**0.298** ± 0.005	0.300 ± 0.007	0.300 ± 0.008	**0.298** ± 0.005	0.299 ± 0.005	0.300 ± 0.009	0.299 ± 0.007	**0.298** ± 0.005
Amazon-Rat.	0.666 ± 0.003	0.642 ± 0.010	0.635 ± 0.006	**0.633** ± 0.006	0.636 ± 0.006	0.637 ± 0.005	0.636 ± 0.004	0.634 ± 0.005
Minesweeper	0.190 ± 0.007	0.180 ± 0.006	0.170 ± 0.014	0.185 ± 0.011	0.181 ± 0.011	0.187 ± 0.005	**0.169** ± 0.014	0.171 ± 0.012
Tolokers	0.278 ± 0.010	**0.268** ± 0.004	**0.268** ± 0.005	0.276 ± 0.012	0.280 ± 0.009	0.277 ± 0.010	0.278 ± 0.009	0.279 ± 0.009
Questions	0.055 ± 0.001	**0.053** ± 0.001	0.054 ± 0.002	**0.053** ± 0.001	**0.053** ± 0.001	**0.053** ± 0.001	**0.053** ± 0.001	**0.053** ± 0.002

levels differ from Table 2 because the ablation is a separate set of runs at one configuration per dataset. This is the behavior the design predicts: the energy score reads the mean logit alone (9), so P can be selected for calibration without affecting detection, and the detection margin comes from the DSS representation under a fixed readout (Section 4.3).

Distribution shift. At the validation-selected standalone configurations of Appendix G, Table 15 varies only P (3 seeds). P=1 is the argmax on both the OOD-validation and the test split on all three settings, so at these configurations the chaos order contributes $+1.7$, $+1.2$, and $+0.6$ points of shifted accuracy over the deterministic P=0 arm.

Table 15: Shifted test accuracy (%) of standalone DSS-GNN on GOOD concept shift as the chaos order varies at the validation-selected configuration (mean ± SD over 3 seeds).

Setting	P=0	P=1	P=2
GOOD-Cora / word	63.24 ± 0.40	**64.89** ± 0.21	64.01 ± 0.25
GOOD-Cora / degree	61.38 ± 0.81	**62.60** ± 0.59	61.22 ± 0.07
GOOD-Arxiv / time	64.81 ± 0.27	**65.43** ± 1.07	64.48 ± 0.27

E.3 Energy-score propagation

Table 16 isolates the score-propagation step of the energy readout on DSS-Hybrid: the raw energy and the propagated energy are computed from the same trained checkpoints (one fixed configuration, 3 runs), so the difference is due to propagation alone. Propagation helps on 9 of 11 settings, by up to $+18.2$ AUROC (Twitch) and $+14.5$ (Cora-structure); the two exceptions are Cora/label (-6.7), where the fixed smoothing averages away part of the leave-out signal inside homophilous neighborhoods, and Amazon-Photo/structure (-0.4), at noise level. The same step transfers to the standalone model (Cora/label $+0.9$, Amazon-Photo/label $+13.4$ at identical checkpoints). We apply propagation uniformly for protocol parity with GNNSafe++.

Table 16: Effect of energy-score propagation on DSS-Hybrid: change in AUROC (points) from the raw to the propagated energy score at identical checkpoints (3 runs, fixed configuration).

Dataset	Structure	Feature	Label
Cora	$+14.5$	$+5.4$	-6.7
Amazon-Photo	-0.4	$+1.7$	$+3.9$
Coauthor-CS	$+1.9$	$+1.1$	$+1.3$
Twitch (cross-graph)		$+18.2$	
Arxiv (cross-graph)		$+7.3$	

E.4 Chaos energy regularization λ_{reg}

Table 17 reports the Brier score as the chaos energy regularization strength λ_{reg} varies from 0 to 0.1 with $P = 2$ fixed. Each column is an independent set of 10-split runs; at each dataset's selected λ_{reg} the cell agrees with the P=2 column of Table 14 within one standard deviation. The

regularizer of (11) penalizes large chaos coefficients; at $\lambda_{\text{reg}} = 0$ the chaos expansion is still active (all $P + 1$ channels are learned), but without an explicit energy penalty. Moderate regularization ($\lambda_{\text{reg}} \in \{10^{-3}, 10^{-2}\}$) yields the best calibration on the majority of datasets. On Texas, λ_{reg} matters most (0.366 at 0 versus 0.213 at 10^{-2}); on Citeseer, Wisconsin, and Roman-Empire λ_{reg}=0 is best and larger values degrade Brier, which is why λ_{reg} is selected per dataset on validation Brier.

Table 17: Effect of chaos energy regularization λ_{reg} on Brier score (↓), with $P = 2$ fixed. Bold indicates the best λ_{reg} per dataset.

Dataset	λ_{reg}=0	10^{-3}	10^{-2}	10^{-1}
Cora	0.219	0.219	**0.207**	0.220
Citeseer	**0.308**	0.310	0.318	0.499
PubMed	0.159	**0.156**	**0.156**	0.157
Texas	0.366	0.246	**0.213**	0.251
Cornell	0.230	0.237	**0.228**	0.342
Wisconsin	**0.103**	0.107	0.108	0.174
Chameleon	0.362	0.354	**0.339**	0.342
Squirrel	0.405	0.407	0.398	**0.393**
CS	0.064	0.066	**0.062**	0.083
Roman-Emp.	**0.298**	0.323	0.603	0.789
Amazon-Rat.	0.663	0.660	**0.633**	**0.633**
Minesweeper	0.186	**0.169**	0.190	0.228
Tolokers	0.281	0.275	**0.268**	0.280
Questions	0.054	**0.053**	0.054	0.055

E.5 Number of quadrature nodes S

Table 18 reports the Brier score as the number of Gauss–Hermite quadrature nodes S varies from 2 to 8, with $P = 2$ fixed. Each column is an independent set of 10-split runs; the S=4 column agrees with the P=2 column of Table 14 within one standard deviation. For the deterministic gates used in these runs ($P_g = 0$), the theoretical minimum for exact chaos arithmetic at $P = 2$ is $S \geq \lceil (P_g + 2P + 1)/2 \rceil = 3$ (Appendix A.1). Brier varies by at most 0.011 across $S \in \{2, \ldots, 8\}$ on 12 of 14 datasets. The exceptions are Wisconsin, where S=2, below the theoretical minimum, gives a higher Brier score (0.142 vs 0.109 at S=3), consistent with the exactness requirement of Theorem 3, and Cornell, where the spread across S (0.218 to 0.286) is within one standard deviation of its 10-split results. The chaos-order sweeps of Tables 13–14 use S=4 throughout, exact for $P \leq 3$; at $P \geq 4$ the highest-order Hermite products are integrated approximately (Appendix D).

Table 18: Brier score (↓) vs. number of quadrature nodes S, with $P = 2$ fixed. The theoretical minimum is $S = 3$.

Dataset	S=2	S=3	S=4	S=6	S=8
Cora	0.216	**0.207**	0.217	0.209	0.212
Citeseer	0.310	**0.308**	0.312	0.313	0.311
PubMed	0.157	**0.156**	**0.156**	0.158	**0.156**
Texas	0.216	0.221	**0.213**	0.217	0.218
Cornell	0.235	0.286	0.228	**0.218**	0.242
Wisconsin	0.142	0.109	0.115	**0.103**	0.108
Chameleon	0.341	0.344	0.343	**0.339**	0.340
Squirrel	0.395	0.397	**0.393**	0.396	0.400
CS	**0.062**	**0.062**	**0.062**	0.063	0.063
Roman-Emp.	**0.298**	**0.298**	**0.298**	**0.298**	**0.298**
Amazon-Rat.	**0.633**	0.634	0.636	0.636	0.637
Minesweeper	0.170	**0.169**	0.170	0.171	0.170
Tolokers	0.269	0.269	0.273	0.270	**0.268**
Questions	0.055	**0.053**	0.054	0.055	0.055

E.6 Contribution decomposition: backbone vs. chaos

Table 19 decomposes the calibration improvement into two factors: the graph-spectral backbone and the chaos expansion. The "ChebNet" column reports a vanilla single-branch Chebyshev filter ($K{=}4$, hidden 64) as a diagnostic backbone baseline, not as part of the main uncertainty-aware baseline comparison in Table 1. The "$P{=}0$" column uses the Table 1 architecture for each dataset (which may include propagate-first or dual branches) with the chaos expansion disabled. The "Best $P{>}0$" column adds the chaos expansion on the same architecture. The $P{=}0 \to P{>}0$ comparison is strictly fair (same architecture, only P changes) and shows that the chaos expansion improves calibration on 12 of 14 datasets (3.6–27.9% Brier reduction), indicating that the stochastic expansion is the primary source of the calibration gains over the $P{=}0$ version in this controlled comparison. The ChebNet column gives useful context: on the small WebKB datasets Texas and Wisconsin, the single-filter ChebNet attains a lower absolute Brier score, while the controlled $P{=}0$ versus $P{>}0$ comparison still isolates the contribution of the chaos expansion. The ChebNet column is also the single-filter control for the filtering design: at $P{=}0$ the per-dataset dual-filter architecture has lower or equal Brier than the single-branch ChebNet on 12 of 14 datasets (equal on Questions; Chameleon 0.362 vs 0.776, Squirrel 0.416 vs 0.798, Roman-Empire 0.298 vs 0.368), with Texas and Wisconsin as the exceptions noted above.

Table 19: Contribution decomposition: Brier score (↓). ChebNet $K{=}4$ is a vanilla single-filter diagnostic; $P{=}0$ and Best $P{>}0$ are the $P{=}0$ column and the best $P{>}0$ column of Table 14 (same per-dataset architecture). Boldface compares the controlled $P{=}0$ versus Best $P{>}0$ pair, and the rightmost column isolates the chaos contribution. Table 1 reports the selected order of Table 12, which can differ from the best swept order.

Dataset	ChebNet $K{=}4$	$P{=}0$	Best $P{>}0$	Δ Chaos
Cora	0.281	0.228	**0.207**	−9.2%
Citeseer	0.470	0.427	**0.308**	−27.9%
PubMed	0.171	0.162	**0.156**	−3.7%
Texas	0.208	0.269	**0.213**	−20.8%
Cornell	0.324	0.241	**0.228**	−5.4%
Wisconsin	0.073	**0.103**	**0.103**	0%
Chameleon	0.776	0.362	**0.339**	−6.4%
Squirrel	0.798	0.416	**0.393**	−5.5%
CS	0.069	0.065	**0.062**	−4.6%
Roman-Emp.	0.368	**0.298**	**0.298**	0%
Amazon-Rat.	0.676	0.666	**0.633**	−5.0%
Minesweeper	0.208	0.190	**0.169**	−11.1%
Tolokers	0.303	0.278	**0.268**	−3.6%
Questions	0.055	0.055	**0.053**	−3.6%

E.7 Computational overhead

Asymptotic complexity. Although the doubly-spectral expansion is defined on the tensor product $\ell^2(V) \otimes L^2(\mathbb{R}, \gamma)$, truncating the chaos basis at order P retains only $P{+}1$ coefficients along the stochastic axis, so the graph-filtering cost and the coefficient memory scale as $(P{+}1)\times$ the deterministic ChebNet baseline and the node-wise dense cost as $S\times$. Concretely, one DSS layer takes $\mathcal{O}\big((P{+}1)(K_{\mathrm{lp}}{+}K_{\mathrm{hp}})|E|d{+}SNd^2{+}S(P{+}1)Nd\big)$ time and $\mathcal{O}((P{+}1{+}S)Nd)$ memory, where $|E|$ is the number of edges, d the hidden width, and S the number of Gauss–Hermite quadrature nodes; the chaos extension is therefore *additive*, not combinatorial, in the stochastic direction, and no eigendecomposition is ever computed. With S at the exactness threshold ($S = P + 1$) the quadrature term is quadratic in P; this is the worst-case operation count, and the measured growth in P is much smaller (below). By Theorem 2 the truncation error decays exponentially in P for targets satisfying the growth condition of Appendix A.6, consistent with the empirical plateau at $P{=}1$–2, so the constant factor over a deterministic backbone stays small. All chaos coefficients share the same Chebyshev filters and projection weights, so the parameter count per layer is $\mathcal{O}\big(K_{\mathrm{lp}} + K_{\mathrm{hp}} + (P{+}1)(P_g{+}1) + d_\ell d_{\ell+1}\big)$ (filter coefficients, gate coefficients $a_{n,r}, b_{n,r}$, and the linear map); the chaos order enters only through the gate coefficients and the input lift. By contrast, the MC-dropout and Deep Ensemble baselines require M forward passes or models at inference ($M{=}20$ and $M{=}5$ in our comparisons).

Wall-clock measurements. Table 20 reports the wall-clock time per training epoch as the chaos order P increases, on three datasets spanning two orders of magnitude in size. The overhead grows linearly with P: each additional chaos order adds one extra graph filter evaluation and Gauss–Hermite projection per layer. At $P = 2$ the per-epoch cost ranges from $2.9\times$ on Cora to $1.2\times$ on Amazon-Ratings and $1.1\times$ on ogbn-arxiv relative to the deterministic baseline ($P = 0$), within the asymptotic bound above and well below the $M\times$ overhead of MC-dropout or ensembles. The measured overhead decreases as graphs grow because the per-order arithmetic is batched into single kernels, and on larger graphs fixed per-epoch costs and better accelerator utilization account for a larger share of the epoch time. On ogbn-arxiv (169K nodes, one H100 GPU) each additional chaos order adds about 4% per-epoch time, and peak GPU memory grows linearly in $P{+}1$ ($1.68/2.75/3.88/5.02$ GiB for $P = 0, \ldots, 3$), consistent with the memory analysis above; peak memory on Cora is 75.9 MiB.

Table 20: Wall-clock time per epoch (ms) vs. chaos order P (Cora and Amazon-Ratings on an A100; ogbn-arxiv on one H100, 3 runs, with peak GPU memory in MiB).

Dataset	$P{=}0$	$P{=}1$	$P{=}2$	$P{=}3$	Ratio $P{=}2/P{=}0$
Cora (2.7K nodes)	35	67	102	131	$2.9\times$
Amazon-Rat. (25K nodes)	464	526	576	646	$1.2\times$
ogbn-arxiv (169K nodes)	2002	2088	2174	2273	$1.1\times$
ogbn-arxiv peak memory (MiB)	1721	2813	3976	5141	$2.3\times$

E.8 Summary of component evidence

Table 21 collects, for each architectural component, the controlled comparison that isolates it and the effect measured. The chaos expansion improves calibration (and, at the selected GOOD configurations, shifted accuracy); the spectral filtering design and the residual improve accuracy under heterophily and shift; the energy score with propagation provides OOD detection. Each task's readout uses a different part of the same representation.

Table 21: Component-wise evidence. Each row names one component, the comparison that isolates it, and the measured effect.

Component	Comparison	Effect
Chaos expansion (P)	$P{=}0$ vs best $P{>}0$, same architecture (Tables 13, 14, 19)	Calibration: Citeseer Brier -27.9% and accuracy $+14.8$ points (the endpoint also beats plain GCN, 0.308 vs 0.318 Brier, Tables 1/23); Texas -20.8%; Minesweeper -11.1%
Chaos expansion (P)	OOD ablation varying only P (Appendix E.2)	Energy AUROC flat across P (Cora-structure $91.0/91.2/90.9/90.8$ at one fixed configuration); detection reads the mean logit, so P can be selected for calibration without changing detection
Chaos expansion (P)	GOOD ablation at the selected configurations (Table 15)	Shifted accuracy: $P{=}1$ beats $P{=}0$ on GOOD-Cora word ($+1.7$), degree ($+1.2$), and GOOD-Arxiv time ($+0.6$)
Spectral backbone (chaos off)	$P{=}0$ vs GCN (Tables 13, 23)	Heterophilous accuracy comes from the spectral backbone: Roman-Empire 79.0 at $P{=}0$ vs GCN 51.3
Filtering design vs single filter	Per-dataset architecture at $P{=}0$ vs single-branch ChebNet (Table 19)	Lower or equal Brier for the full design on 12 of 14 datasets (equal on Questions; Chameleon 0.362 vs 0.776, Squirrel 0.416 vs 0.798, Roman-Empire 0.298 vs 0.368); exceptions are Texas and Wisconsin
DSS residual branch (as a unit)	Hybrid vs identically trained GCN base, 14 datasets (Table 24)	Accuracy: Roman-Empire $+27.7$, Minesweeper $+6.6$; no dataset loses beyond noise (worst -0.8)
DSS representation under a fixed OOD readout	DSS-Hybrid vs GNNSafe++ (same energy readout, propagation, objective, GCN encoder in both; Table 2)	Cora $94.3/97.6/94.1$ vs $90.6/95.6/92.8$
Energy-score propagation	Same-checkpoint on/off, all 11 settings (Table 16)	Helps on 9 of 11, up to $+18.2$ AUROC (Twitch) and $+14.5$ (Cora-structure)

F Comparison with Standard GNN Baselines

Table 1 in the main text compares DSS-GNN against recent uncertainty-aware graph methods (TFE-GNN and G-ΔUQ). Here we additionally compare against GCN [15] and GAT [34], two widely used GNN architectures that do not incorporate any uncertainty mechanism. Both baselines use 2 layers, 64 hidden units, Adam optimizer (lr=0.01, weight decay=5×10^{-4}), and the same 10 splits as DSS-GNN.

Table 23 reports accuracy and Brier score. The main observation is that *higher accuracy does not imply better calibration*. On Cora, GCN and GAT both achieve higher accuracy than DSS-GNN (87.5% and 87.8% vs. 86.6%) yet have worse Brier scores (0.213 and 0.222 vs. 0.207). On Citeseer, GAT matches DSS-GNN in accuracy (80.6% vs. 80.2%) but is substantially worse in Brier (0.353 vs. 0.308). On many heterophilous benchmarks, GCN and GAT have much lower accuracy and higher Brier scores than DSS-GNN. The calibration gains of DSS-GNN are therefore not an accuracy effect.

Stochastic baselines under the same protocol. Table 22 adds MC-dropout ($M=20$ stochastic passes at test time, probabilities averaged) and Deep Ensembles ($M=5$ independently trained models, probabilities averaged) on the GCN of Table 23, under the full 10-split protocol and the same code path; the plain-GCN control from that code path reproduces Table 23 (Cora 87.47 vs 87.45, Wisconsin 50.25 vs 50.25). DSS-GNN's Brier is lower on all 14 datasets than either baseline, and ensembling improves the GCN's Brier by at most 0.008 on any dataset while MC-dropout sometimes worsens it (Cora): averaging a conventional GNN does not remove the calibration difference. On Cora and Questions the margins are small (0.207 vs 0.213 and 0.053 vs 0.055). The same multi-pass cost structure applies to sampling-based Bayesian GNNs and latent-variable GNNs; the chaos expansion is instead deterministic, with no sampling variance.

Table 22: Brier score (↓) with test accuracy (%) in parentheses for DSS-GNN (Table 1) and stochastic GCN baselines under the same 10-split protocol (means over 10 splits).

Dataset	DSS-GNN	MC-dropout	Deep Ensemble	Plain GCN
Cora	**0.207** (86.63)	0.2188 (87.36)	0.2134 (87.45)	0.2125 (87.47)
Citeseer	**0.308** (80.20)	0.3230 (80.03)	0.3170 (79.77)	0.3176 (80.00)
PubMed	**0.156** (89.72)	0.2176 (86.20)	0.2153 (86.16)	0.2155 (86.10)
Texas	**0.240** (91.31)	0.5862 (64.43)	0.5686 (62.79)	0.5758 (64.43)
Cornell	**0.231** (85.11)	0.6829 (52.77)	0.6826 (52.98)	0.6839 (52.34)
Wisconsin	**0.104** (93.25)	0.6370 (50.38)	0.6373 (51.25)	0.6435 (50.25)
Chameleon	**0.339** (75.36)	0.7378 (41.73)	0.7371 (42.71)	0.7374 (42.21)
Squirrel	**0.394** (68.06)	0.7850 (28.11)	0.7847 (28.81)	0.7849 (28.38)
CS	**0.062** (96.17)	0.0821 (94.55)	0.0809 (94.63)	0.0816 (94.58)
Roman-Emp.	**0.300** (78.84)	0.6554 (50.92)	0.6471 (51.49)	0.6477 (51.37)
Amz-Rat.	**0.634** (49.72)	0.6772 (44.08)	0.6769 (44.00)	0.6771 (44.07)
Minesweeper	**0.169** (87.66)	0.2874 (80.20)	0.2874 (80.20)	0.2874 (80.21)
Tolokers	**0.268** (79.88)	0.2970 (78.78)	0.2970 (78.78)	0.2971 (78.79)
Questions	**0.053** (97.17)	0.0548 (97.06)	0.0549 (97.07)	0.0549 (97.07)

Table 23: Node classification: test accuracy (%, ↑) and Brier score (↓) for DSS-GNN vs. standard GNN baselines. Mean ± SD over 10 splits. Best per dataset in **bold**.

	Accuracy (%)			Brier Score		
Dataset	DSS-GNN	GCN	GAT	DSS-GNN	GCN	GAT
Cora	86.63 ± 1.26	87.45 ± 1.61	**87.75** ± 1.71	**0.207** ± 0.019	0.213 ± 0.014	0.222 ± 0.015
Citeseer	80.20 ± 1.28	79.84 ± 0.96	**80.57** ± 1.12	**0.308** ± 0.010	0.318 ± 0.007	0.353 ± 0.007
PubMed	**89.72** ± 0.31	86.21 ± 0.31	85.72 ± 0.28	**0.156** ± 0.005	0.215 ± 0.005	0.232 ± 0.004
Texas	**91.31** ± 3.44	64.43 ± 6.80	71.80 ± 7.06	**0.240** ± 0.111	0.576 ± 0.031	0.555 ± 0.038
Cornell	**85.11** ± 5.12	52.34 ± 9.99	49.57 ± 12.63	**0.231** ± 0.079	0.684 ± 0.080	0.631 ± 0.073
Wisconsin	**93.25** ± 3.39	50.25 ± 5.30	56.62 ± 4.71	**0.104** ± 0.047	0.643 ± 0.024	0.598 ± 0.021
Chameleon	**75.36** ± 1.19	42.21 ± 1.78	43.83 ± 2.54	**0.339** ± 0.019	0.737 ± 0.009	0.712 ± 0.015
Squirrel	**68.06** ± 2.21	28.39 ± 1.17	27.57 ± 0.75	**0.394** ± 0.027	0.785 ± 0.003	0.789 ± 0.002
CS	**96.17** ± 0.23	94.61 ± 0.24	94.45 ± 0.18	**0.062** ± 0.004	0.082 ± 0.004	0.085 ± 0.004
Roman-Emp.	**78.84** ± 0.61	51.30 ± 0.31	42.85 ± 0.53	**0.300** ± 0.007	0.648 ± 0.001	0.755 ± 0.003
Amz-Rat.	**49.72** ± 0.64	43.97 ± 0.37	42.20 ± 0.43	**0.634** ± 0.005	0.677 ± 0.002	0.690 ± 0.001
Minesweeper	**87.66** ± 1.10	80.21 ± 0.19	80.10 ± 0.08	**0.169** ± 0.014	0.287 ± 0.003	0.281 ± 0.003
Tolokers	**79.88** ± 0.49	78.81 ± 0.47	78.19 ± 0.16	**0.268** ± 0.005	0.297 ± 0.005	0.299 ± 0.002
Questions	**97.17** ± 0.04	97.08 ± 0.02	97.02 ± 0.00	**0.053** ± 0.001	0.055 ± 0.000	0.057 ± 0.000

G Cross-Evaluation of the Two Deployment Modes

This appendix gives the full tables for Section 4.5: the hybrid under the calibration protocol, the standalone model under the OOD and GOOD protocols, the readout-coincidence measurement of Section 4.2, the effect of BatchNorm under the calibration protocol, and the covariate-shift counterpart of Table 4.

Table 5 (Section 4.5) summarizes both deployment modes on all three tasks; the per-dataset comparisons summarized in its cells are Tables 24, 25, and 26 below.

G.1 Protocols

Hybrid under the calibration protocol. DSS-Hybrid (2-layer GCN base with the DSS residual branch of Appendix D) is trained under the identical 10-split protocol of Table 1, paired with its own GCN base trained identically with the residual removed and matched budgets, so the pair isolates the residual branch.

Standalone under the OOD protocol. Standalone DSS-GNN is trained in the GNNSafe pipeline (200 epochs, 3 runs, energy score with propagation, pipeline defaults: hidden 64, 2 layers, weight decay 10^{-2}, learning rate 10^{-2}, P=1, S=4). Rows marked +BN add the per-chaos-channel BatchNorm of Appendix D, with every other hyperparameter unchanged; the variant is selected on the validation split. Twitch uses learning rate 10^{-3}, hidden 128, and no BatchNorm, selected on validation ID accuracy. The Arxiv row is trained with the GNNSafe++ energy-margin objective at the Arxiv margin hyperparameters of Wu et al. [36] with $m_{\text{out}} = -1$, selected on validation ID accuracy. All other rows use standard ID training.

Standalone under the GOOD protocol. Standalone DSS-GNN is trained in the GOOD pipeline with L_1 row-normalized input features, 2 layers, K_{lp}=3, K_{hp}=2, 500 epochs, patience 200, 3 seeds; the epoch is selected by OOD-validation loss within each run and the configuration (BatchNorm on/off, learning rate, hidden width, P) on the GOOD OOD-validation split: CBAS and WebKB +BN, learning rate 10^{-2}; Twitch +BN, 10^{-3}; GOOD-Cora/word +BN, 5×10^{-3}, hidden 128, weight decay 10^{-3}; GOOD-Cora/degree +BN, 3×10^{-3}, hidden 128; GOOD-Arxiv/time +BN, 10^{-2}, hidden 256; GOOD-Arxiv/degree no BatchNorm, 10^{-2}, hidden 64; P=1 throughout.

G.2 Calibration: standalone, hybrid, and the hybrid's GCN base

Table 24: Test accuracy (%) and Brier score, mean ± SD over the same 10 splits. The standalone column is Table 1; the hybrid and its GCN base are a matched pair (identical training, budgets, and splits; the only difference is the DSS residual).

	Standalone DSS-GNN		DSS-Hybrid		GCN base	
Dataset	Accuracy	Brier	Accuracy	Brier	Accuracy	Brier
Cora	86.63 ± 1.26	0.207 ± 0.019	84.33 ± 0.97	0.241 ± 0.015	84.20 ± 1.02	0.243 ± 0.016
Citeseer	80.20 ± 1.28	0.308 ± 0.010	74.71 ± 1.05	0.375 ± 0.009	74.32 ± 1.13	0.378 ± 0.009
PubMed	89.72 ± 0.31	0.156 ± 0.005	88.27 ± 0.56	0.177 ± 0.006	87.93 ± 0.56	0.181 ± 0.006
Texas	91.31 ± 3.44	0.240 ± 0.111	45.57 ± 7.68	0.737 ± 0.034	46.39 ± 7.55	0.737 ± 0.034
Cornell	85.11 ± 5.12	0.231 ± 0.079	55.11 ± 14.03	0.656 ± 0.142	47.66 ± 12.01	0.727 ± 0.068
Wisconsin	93.25 ± 3.39	0.104 ± 0.047	48.12 ± 6.50	0.680 ± 0.029	48.50 ± 6.14	0.680 ± 0.028
Chameleon	75.36 ± 1.19	0.339 ± 0.019	33.50 ± 2.10	0.777 ± 0.004	34.00 ± 2.44	0.777 ± 0.004
Squirrel	68.06 ± 2.21	0.394 ± 0.027	26.35 ± 2.02	0.790 ± 0.003	26.23 ± 2.36	0.790 ± 0.003
CS	96.17 ± 0.23	0.062 ± 0.004	94.26 ± 0.24	0.087 ± 0.004	94.23 ± 0.21	0.088 ± 0.003
Roman-Empire	78.84 ± 0.61	0.300 ± 0.007	74.36 ± 0.61	0.361 ± 0.007	46.63 ± 0.26	0.690 ± 0.003
Amazon-Ratings	49.72 ± 0.64	0.634 ± 0.005	46.36 ± 0.57	0.661 ± 0.004	46.32 ± 0.48	0.661 ± 0.002
Minesweeper	87.66 ± 1.10	0.169 ± 0.014	86.91 ± 0.45	0.184 ± 0.006	80.31 ± 0.21	0.289 ± 0.003
Tolokers	79.88 ± 0.49	0.268 ± 0.005	81.19 ± 0.50	0.255 ± 0.006	79.46 ± 0.34	0.291 ± 0.006
Questions	97.17 ± 0.04	0.053 ± 0.001	97.09 ± 0.04	0.053 ± 0.001	97.08 ± 0.04	0.054 ± 0.001

The hybrid is at or above its own base on 11 of 14 datasets in accuracy; the three negative deltas (Texas -0.82, Chameleon -0.50, Wisconsin -0.38) are far inside one standard deviation, and every positive Brier delta (hybrid worse) is at most 0.0008, also far inside one standard deviation. The gains are structural and large where the GCN inductive bias fails: Roman-Empire $+27.7$ accuracy and

-0.33 Brier, Minesweeper $+6.6$ and -0.11, Tolokers $+1.7$ and -0.04. On the small heterophilous graphs where the GCN backbone itself fails (Texas, Cornell, Wisconsin, Chameleon, Squirrel), the residual matches the base rather than improving on it, and the standalone model is the deployment choice. In hybrid mode the argmax of the quadrature predictive and of the mean logit agree on every test node of every dataset (disagreement 0.0000 on all 14).

G.3 OOD detection: standalone versus hybrid

Table 25: AUROC (%) and ID test accuracy (%) on the 11 OOD settings. Standalone: 3 runs, mean $\pm$ SD; hybrid: Tables 2–3 (AUROC only for the nine node-OOD settings*). The last column gives the standalone training objective.

Setting	Standalone AUROC	Standalone ID acc.	Hybrid AUROC	Hybrid ID acc.	Objective (standalone)
Cora / structure	79.79 $\pm$ 1.46 (+BN)	69.50	94.32	n/r*	standard (no OOD exposure)
Cora / feature	88.15 $\pm$ 1.02 (+BN)	69.17	97.60	n/r*	standard (no OOD exposure)
Cora / label	94.03 $\pm$ 1.09 (+BN)	89.24	94.11	n/r*	standard (no OOD exposure)
Amazon-Photo / structure	96.53 $\pm$ 1.41 (+BN)	91.59	99.69	n/r*	standard (no OOD exposure)
Amazon-Photo / feature	98.10 $\pm$ 0.11 (+BN)	92.00	99.66	n/r*	standard (no OOD exposure)
Amazon-Photo / label	96.53 $\pm$ 0.10 (+BN)	94.81	97.52	n/r*	standard (no OOD exposure)
Coauthor-CS / structure	94.33 $\pm$ 0.29 (+BN)	92.64	99.96	n/r*	standard (no OOD exposure)
Coauthor-CS / feature	97.79 $\pm$ 0.16 (+BN)	92.47	99.95	n/r*	standard (no OOD exposure)
Coauthor-CS / label	95.87 $\pm$ 1.83 (+BN)	94.00	98.07	n/r*	standard (no OOD exposure)
Twitch (cross-graph)	51.85 $\pm$ 4.83 (no separation)	68.39 $\pm$ 0.20	95.75	70.52	standard (no OOD exposure)
Arxiv (cross-graph)	73.73 $\pm$ 0.24 (+BN)	61.16 $\pm$ 0.58	73.36	53.99	GNNSafe++ margin objective

*Table 2 reports AUROC only. Hybrid ID accuracy on these nine settings (3 runs under the same protocol) is Cora $81.6/81.1/90.6$, Amazon-Photo $92.4/91.0/95.4$, Coauthor-CS $93.0/93.0/95.5$ (structure/feature/label).

The hybrid is the stronger detector on 10 of 11 settings; the standalone trains on all 11 (ID accuracies in Table 25) and is competitive on the label-shift settings (Cora/label 94.03 vs 94.11) and on Arxiv, where under the GNNSafe++ margin objective it reaches 73.73 AUROC, above the hybrid's 73.36, with the best Arxiv ID accuracy of any method in Table 3 (61.16). The one standalone exception is Twitch: the classifier itself trains (ID accuracy 68.39 vs the hybrid's 70.52), but its energy score does not separate the cross-graph OOD inputs, so the hybrid (95.75) is preferred there. The per-chaos-channel BatchNorm of Appendix D is used on all nine perturbation settings, where the sparse public splits require it for every backbone in the pipeline (Section 4.5).

G.4 Distribution shift: standalone versus hybrid

Table 26: Shifted (OOD) test accuracy (%) on the GOOD concept-shift settings. Standalone: mean $\pm$ SD over 3 seeds, configuration selected on the GOOD OOD-validation split (Appendix G.1); hybrid and best baseline from Table 4.

Setting	Standalone	Hybrid	Best baseline
GOOD-CBAS / color	82.38 $\pm$ 13.17 (+BN)	88.57	TAR 87.29
GOOD-WebKB / university	36.41 $\pm$ 1.60 (+BN)	40.77	TAR 30.83
GOOD-Twitch / language	60.28 $\pm$ 0.51 (+BN)	61.21	TAR 57.20
GOOD-Cora / word	64.89 $\pm$ 0.21 (+BN)	64.81	TAR 64.73
GOOD-Cora / degree	62.60 $\pm$ 0.59 (+BN)	62.73	TAR 61.73
GOOD-Arxiv / time	65.43 $\pm$ 1.07 (+BN)	66.76	TAR 66.08
GOOD-Arxiv / degree	64.62 $\pm$ 0.15	66.43	G-ΔUQ 64.34

The standalone model is above every baseline of Table 4 on 5 of 7 settings (WebKB, Twitch, Cora word, Cora degree, Arxiv degree) and at ERM's level on the remaining two (ERM: 82.43 on CBAS and 65.64 on Arxiv/time), while the hybrid is above every baseline on all 7; on GOOD-Cora/word the two modes agree within noise (64.81 vs 64.89 ± 0.21). GOOD-CBAS is a small synthetic graph with the largest seed variance of the seven settings (standalone SD 13.2 over 3 seeds). ID accuracies of the standalone runs are between 63.5 and 96.2 across the seven settings (CBAS 96.2, WebKB 76.0, Twitch 63.5, Cora word 68.8, Cora degree 68.9, Arxiv time 74.1, Arxiv degree 71.5).

G.5 Readout coincidence

Table 27 compares the two readouts of Section 3.2 at identical checkpoints.

Table 27: Quadrature readout versus mean-logit readout at identical checkpoints under the full 10-split protocol of Table 1: absolute Brier difference and the fraction of test nodes on which $\arg\max \bar{p}_i$ and $\arg\max Z_{i,0}$ disagree (mean over 10 splits). Accuracy differs by at most 0.04 points on every dataset. In hybrid mode the disagreement is 0.0000 on all 14 datasets.

Dataset	$\lvert\Delta$Brier$\rvert$	Disagree (%)	Dataset	$\lvert\Delta$Brier$\rvert$	Disagree (%)
Cora	0.0002	0.03	Roman-Emp.	0.0000	0.00
Citeseer	0.0000	0.00	Amz-Rat.	0.0000	0.00
PubMed	0.0000	0.02	Minesweeper	0.0003	0.21
Texas	0.0021	0.00	Tolokers	0.0000	0.00
Cornell	0.0000	0.00	Questions	0.0000	0.00
Wisconsin	0.0000	0.00	CS	0.0001	0.00
Chameleon	0.0001	0.00	Squirrel	0.0007	0.08

G.6 BatchNorm under the calibration protocol

Table 28 re-runs the Table 1 protocol with and without the per-chaos-channel BatchNorm on four datasets (10 splits each). The BatchNorm-free controls reproduce Table 1 within a fraction of one standard deviation (the Citeseer run uses the selected $P{=}2$ configuration of Table 12, with $\lambda_{\mathrm{reg}}{=}0$, and reproduces that row as well). With BatchNorm, Cora and Roman-Empire are unchanged within noise and Wisconsin is within one standard deviation, while Citeseer degrades far beyond noise (-13.2 accuracy). BatchNorm is therefore a stabilization for the sparse-label OOD pipeline, where every baseline backbone also uses it, and is not used under the full-label calibration protocol.

Table 28: Test accuracy (%) / Brier, mean $\pm$ SD over 10 splits, under the protocol of Table 1 at one calibration configuration per dataset, without and with per-chaos-channel BatchNorm.

Dataset	BatchNorm off (control)	Table 1	BatchNorm on
Cora	86.44 ± 1.56 / 0.216 ± 0.022	86.63 ± 1.26 / 0.207 ± 0.019	85.78 ± 1.76 / 0.220 ± 0.017
Citeseer	80.27 ± 1.18 / 0.307 ± 0.010	80.20 ± 1.28 / 0.308 ± 0.010	67.03 ± 1.85 / 0.480 ± 0.024
Wisconsin	93.62 ± 1.72 / 0.109 ± 0.029	93.25 ± 3.39 / 0.104 ± 0.047	91.75 ± 4.75 / 0.134 ± 0.061
Roman-Empire	78.82 ± 1.08 / 0.300 ± 0.012	78.84 ± 0.61 / 0.300 ± 0.007	78.60 ± 0.64 / 0.300 ± 0.009

G.7 Covariate shift on GOOD

Table 29 is the covariate-shift counterpart of Table 4: the same seven GOOD settings under their covariate splits, comparing ERM-trained DSS-Hybrid (GCN encoder, configuration selected by validation loss) with the same-protocol ERM GCN of the GNNSafe pipeline (3 seeds). DSS-Hybrid is above the ERM control on 6 of 7 settings; the one loss, GOOD-Cora/degree, is within 1.8 points. The two Arxiv ERM baselines did not converge within the shared 120-epoch budget (ID accuracy 45 to 50, versus 72 to 77 for the DSS runs on the same splits), so those two margins are not informative on their own. With the seven concept-shift settings of Table 4 this covers both shift families that GOOD defines.

Table 29: Shifted test accuracy (%) under GOOD covariate shift: ERM-trained DSS-Hybrid (GCN encoder, validation-loss-selected configuration) versus the same-protocol ERM GCN. *ERM baseline not converged within the shared budget (see text).

Setting (covariate split)	ERM GCN	DSS-Hybrid (ERM)
GOOD-CBAS / color	49.05	**53.33**
GOOD-WebKB / university	13.49	**29.10**
GOOD-Twitch / language	42.78	**52.64**
GOOD-Cora / word	61.84	**64.18**
GOOD-Cora / degree	**56.66**	54.94
GOOD-Arxiv / time	44.80*	**70.39**
GOOD-Arxiv / degree	25.68*	**57.60**

H Architecture Overview and Notation

Figure 4 gives a schematic of the DSS-GNN architecture of Section 3: the two spectral axes that index the chaos coefficients, one DSS layer, the two readouts and the chaos energy, and the two deployment modes. Table 30 summarizes the notation of the DSS layer, Eqs. (2)–(6).

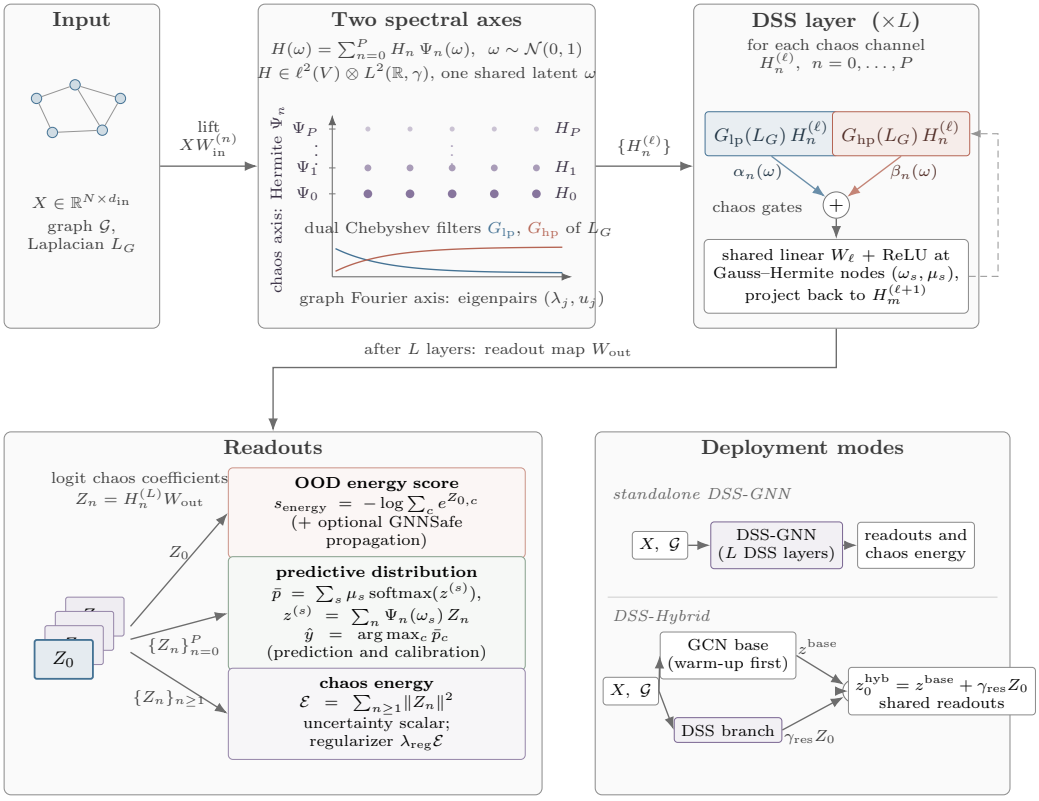

Figure 4: Overview of DSS-GNN. Input features X on graph $\mathcal{G}$ are lifted to chaos coefficients $H_n^{(0)} = XW_{\text{in}}^{(n)}$, which are indexed by two spectral axes: the graph Fourier axis (Laplacian eigenpairs (λ_j, u_j), filtered by dual Chebyshev filters $G_{\text{lp}}, G_{\text{hp}}$) and the Wiener chaos axis (Hermite basis Ψ_n of a scalar Gaussian latent ω). Each DSS layer propagates every chaos channel $H_n^{(\ell)}$ through both filters, couples them with per-order chaos gates α_n, β_n, and applies a shared linear map and ReLU at Gauss–Hermite nodes before projecting back (Eqs. (3)–(6)). The final coefficients $Z_n = H_n^{(L)} W_{\text{out}}$ are the inputs of the two task readouts, the mean-logit energy score for OOD detection (9) and the quadrature-averaged predictive $\bar{p}$ for class prediction and calibration (8), together with the chaos energy $\mathcal{E}$, which is used as regularizer and uncertainty scalar (7) (bottom left). Bottom right: the two deployment modes, standalone DSS-GNN and DSS-Hybrid, which adds the DSS branch residually (scale γ_{res}) to a warmed-up GCN base (Section 3.4); both use the same readouts. The node index i is suppressed for readability.

Table 30: Notation for the DSS layer, Eqs. (2)–(6).

Symbol	Meaning
ω	the single shared standard Gaussian latent; never sampled, integrated out by quadrature
Ψ_n	normalized Hermite polynomial of order n; orthonormal basis of $L^2(\mathbb{R}, \gamma)$
P	chaos truncation order; $n, m = 0, \ldots, P$ index input/output chaos orders
$H_n^{(\ell)} \in \mathbb{R}^{N \times d_\ell}$	order-n chaos coefficient of the layer-ℓ embeddings (2); the embedding field is $H^{(\ell)}(\omega) = \sum_n H_n^{(\ell)} \Psi_n(\omega)$
$\widetilde{L}_G, T_k$	rescaled Laplacian $2L_G/\lambda_{\max} - I$ and Chebyshev polynomial of degree k (Section 2)
$G_{\mathrm{lp}}, G_{\mathrm{hp}}$	learned low/high-pass filter polynomials of degrees $K_{\mathrm{lp}}, K_{\mathrm{hp}}$ with coefficients $c_k^{(\mathrm{lp})}, c_k^{(\mathrm{hp})}$ (3)
$\alpha_n(\omega), \beta_n(\omega)$	per-order chaos gates coupling the two filter branches; deterministic scalars, or order-P_g expansions with coefficients $a_{n,r}, b_{n,r}$ (4)
ω_s, μ_s, S	Gauss–Hermite quadrature nodes and weights, $s = 1, \ldots, S$; exact for the filter/gate polynomials when $S \geq P + 1$ (for $P_g \leq 1$)
$\widetilde{H}^{(\ell+1)}(\omega_s)$	pre-activation field evaluated at quadrature node ω_s (5)
W_ℓ, σ	trainable linear map $d_\ell \to d_{\ell+1}$ and pointwise activation; Eq. (6) projects $\sigma(\widetilde{H}^{(\ell+1)}(\omega_s))$ back onto the Ψ_m basis

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the contributions, which Sections 3 and 4 support, and the limitations are stated in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 discusses three limitations (two deployment modes with their regimes, the scalar latent, and compute), with the supporting measurements in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: every theorem, proposition, and corollary states its assumptions and is proved in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendix D gives the training protocol and the per-dataset configurations (Tables 11 and 12), and Appendix E reports the sensitivity to the main hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: the code, configuration files, and instructions for reproducing every table are available at `https://github.com/heraclixus/DSSGNN`; dataset sources and splits are documented in Appendix C.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: splits, optimizers, hyperparameter grids, and the per-dataset selected configurations are given in Appendix D (Tables 11 and 12).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Calibration results (Table 1 and the chaos-order sweeps of Appendix E) report mean $\pm$ standard deviation over 10 splits; the cross-evaluation runs on OOD detection and GOOD (3 seeds) report standard deviations in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: GPU types and the compute budget are given in Appendix D; per-epoch time and peak memory are reported in Table 20 (Appendix E).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public benchmark graphs and involves no human subjects or personal data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is a methodological contribution on uncertainty estimation for node classification, evaluated on public benchmarks; we do not identify a direct path to negative societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases no pretrained models or scraped datasets that carry a risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and baseline implementations are cited, and their sources and licenses are listed in Appendix C.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented in its repository (installation, data preparation, and the command for each table).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work involves neither crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work involves neither crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not involve LLMs as an important, original, or non-standard component.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.