GEOMETRY-INDUCED DIFFUSION ON GRAPHS: LEARNING TASK-ADAPTIVE PROPAGATION FOR GNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-range information propagation remains challenging in Graph Neural Networks (GNNs), where repeated local aggregation can lead to oversmoothing and oversquashing. Existing approaches often rely on global attention mechanisms or modify the graph topology through rewiring, potentially increasing computational or architectural complexity. We introduce a learnable node-wise function μ that induces a weighted graph Laplacian L_μ, allowing the propagation geometry to adapt to the task while preserving the original graph topology. This construction is not specific to a particular GNN architecture, but we instantiate it within Chebyshev spectral convolutions, yielding a novel μ-ChebNet layer. We show that learning μ reshapes the spectrum of the diffusion operator, modifying how long different graph signal components persist during propagation and thus the timescale over which oversmoothing occurs. On a controlled bottleneck example, we further show that reweighting the graph through μ can reduce effective resistance, a quantity previously connected to oversquashing. Experiments on synthetic long-range tasks and real-world graph benchmarks show improved or competitive predictive performance while retaining the computational structure of sparse polynomial graph filtering.

1 INTRODUCTION

Graph Neural Networks (GNNs) (Scarselli et al., 2008; Bruna et al., 2013; Defferrard et al., 2016) have achieved significant success in recent years, driving the development of novel architectures and investigations into their theoretical foundations. Alongside these advances, important limitations have also come to light, most notably oversmoothing and oversquashing (Shao et al., 2023; Rusch et al., 2023; Jin & Zhu, 2025; Attali et al., 2024). Oversmoothing makes node representations increasingly indistinguishable, while oversquashing compresses information from many distant nodes into limited-size embeddings. These effects are particularly damaging for long-range tasks, where task-relevant information must propagate across many steps and may progressively weaken (Arroyo et al., 2025).

These limitations motivate to take a closer look on how information propagates through the graph. GNN layers can be interpreted through the lens of continuous dynamics, by considering feature propagation as the discretization of a dynamical evolution on the graph (Poli et al., 2019). It has motivated diffusion-based formulations that recast message passing as the discretization of partial differential equations (PDEs) on graphs, enabling principled control over information propagation through learnable diffusivity (Chamberlain et al., 2021a) or global attention mechanisms (Wu et al., 2023a; 2025a). Although the continuous GNN diffusion perspective provides a principled foundation for designing expressive architectures, it often leads to high computational complexity due to discretization.

A complementary approach acts on the graph geometry through rewiring, for example by adding task-dependent edges or modifying the connectivity structure (Gutteridge et al., 2023; Barbero et al., 2024). While effective, such methods can introduce additional preprocessing cost, alter the original edge set and they may change the semantics of the input graph.

Conversely, spectral GNNs (Defferrard et al., 2016; Kipf & Welling, 2017) offer simple and computationally efficient architectures by leveraging the Fourier domain of the graph, only requiring to learn the fixed spectral filters' weights. Recent work on stabilization of the information flow

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

across layers (Gravina et al., 2023; Hariri et al., 2026) has shown that stabilized spectral GNNs can outperform most other GNNs on long-range tasks, and it is believed that they possess competitive generalization qualities (Levie et al., 2021). Such spectral GNNs, however, still rely on a *fixed* graph Laplacian L, limiting their ability to adapt information flow to task-specific structure. In particular, they lack a mechanism for locally steering information propagation.

Our objective is therefore to shape information diffusion on graphs in a task-dependent and data-driven manner, producing a rewiring-like effect, while preserving the original graph topology. To this end, we learn a node-wise density μ, which induces a weighted Laplacian L_μ by reweighting existing edges according to the densities at their endpoints. This is analogous to diffusion in an inhomogeneous medium, where spatial variations in the medium locally guide propagation. To characterize how μ affects diffusion, we derive mode-dependent spectral bounds for L_μ (Theorem 3) that quantify how μ can modify the eigenvalues associated with different eigenspaces. These results provide the foundation for a modal analysis of finite-time diffusion. It shows that the learned geometry can modify the relative diffusion timescales of different spectral components, allowing some modes to persist longer relative to others and this can provide a mechanism for delaying oversmoothing. We also show that μ can reduce oversquashing, by diminishing the effective resistance across graph bottlenecks, without modifying the graph topology. We then extend the analysis to study the effects of nonlinear activations. Under suitable regularity assumptions, Theorem 4 gives control of signal evolution through graph neural network layers. Motivated by these theoretical insights, we introduce μ-ChebNet, a lightweight spectral GNN in which the diffusion geometry is learned directly from data. Experiments on synthetic and real-world benchmarks demonstrate that this adaptive diffusion mechanism provides a competitive alternative to more complex architectures for tasks involving long-range dependencies, while retaining interpretability and computational simplicity.

Our work makes the following contributions:

1. **A task-adaptive diffusion geometry for GNNs.** We derive a discrete Laplacian L_μ whose entries depend on a learnable node-wise density μ. This replaces the fixed graph Laplacian with a learned propagation operator that preserves the original topology.

2. **Spectral control of propagation dynamics.** We characterize how μ reshapes spectral decay and effective resistance, connecting the learned geometry to oversmoothing and oversquashing. We further derive bounds on how nonlinear activations affect this mode-selective propagation.

3. **A new family of spectral GNNs.** We introduce μ-ChebNet an architecture that jointly learns (i) propagation dynamics through the density μ and (ii) spectral filters through Chebyshev polynomials. The model bridges spectral and message-passing viewpoints: it retains the efficiency of polynomial spectral filtering, while propagation remains a local diffusion process along existing edges, now governed by a learned task-adaptive Laplacian rather than the fixed graph Laplacian.

4. **Empirical evidence and interpretability.** On controlled long-range tasks, μ-ChebNet achieves strong performance already at shallow depths, indicating that adapting the propagation geometry can reduce the number of layers needed for effective long-range information transfer. On real-world benchmarks, the model remains competitive with strong spectral and transformer baselines. The learned node-wise density μ additionally provides an interpretable graph signal, revealing where propagation is strengthened or attenuated in a given task.

Outline. We review calculus on graphs in Section 3.1, leading to the key anisotropic diffusion equation in Section 3.2. We construct said anisotropic diffusion on graphs through a weighted graph Laplacian L_μ in Section 4.1, and prove general beneficial spectral properties in Section 4.2, supplemented by pedagogical examples. Leveraging these newfound results, we propose μ-ChebNet as a spectral GNN implementing a task-dependent, learnable L_μ in Section 5. Experimental results on both synthetic and real-world tasks are reported in Section 6, illustrating how the learned L_μ provides a simple yet interpretable mechanism for task-dependent routing.

2 Related Work

Mitigating Oversmoothing & Oversquashing. Recent works use PDE-inspired propagation dynamics to control information flow though diffusion and transport mechanisms. GRAND casts GNN layers as discretizations of diffusion equations on graphs, where the choice of diffusivity and numerical scheme shapes propagation (Chamberlain et al., 2021b). Wu et al. (2025b) extend this view through advective diffusion: global attention is interpreted as a non-local diffusion mechanism that captures latent interactions, while local message passing acts as an advection-like term that keeps the dynamics tied to the observed graph topology. While expressive and theoretically motivated, such approaches typically rely on attention-based coupling or the solution of continuous-time dynamics, which can introduce substantial computational overhead. Graph Transformers (GTs) (Rampášek et al., 2022) pursue a related goal from an architectural perspective, combining local aggregation with global attention to improve long-range communication. Yet, standard global attention scales quadratically in the number of nodes, motivating sparse GTs (Shirzad et al., 2023; 2024; Chen et al., 2022) and dynamical graph-rewiring schemes (Arnaiz-Rodríguez et al., 2022).

Efficient Spectral GNNs. Spectral GNNs (such as ChebNets (Defferrard et al., 2016)) have undergone a recent revival in relevance for both their computational efficiency and ability to capture long-range interactions via constructive graph Laplacian-based filters when stabilized (Gravina et al., 2023). These models are simple to implement and analyze due to the rich tools of spectral graph theory (Chung, 1997), and were also shown to consistently outperform MPNNs, graph rewiring techniques, and GTs on several long-range tasks (Hariri et al., 2026).

Modified Graph Laplacians. Rather than designing new architectures, there has been considerable research done in redefining the graph Laplacians, with hopes of enriching information propagation. Relying on graph signal processing, Girault et al. (2018) introduce vertex-wise weights through a modified inner product, adapting the graph Fourier representation to irregular graph structures. Modern GNNs-based approaches include degree-dependent penalties (Gong et al., 2015), attractive & repulsive effects modeling (Li et al., 2023), complex phases for orientation (Zhang et al., 2021; Fiorini et al., 2023), or control over spectral scaling (Maskey et al., 2023). In contrast, our adaptive Laplacian L_μ learns node-wise weights directly from the downstream task, yielding spatially varying, task-dependent transport while preserving the original graph topology. Rather than shortening paths by adding edges, this changes the effective transport geometry on the existing graph.

3 Background

In this section we recall elements of calculus on graphs, which will allow us to formulate advection-diffusion dynamics on graphs and derive both the classic notion of a graph Laplacian L, as well as the task-adaptive weighted Laplacian L_μ. We will also briefly recall the ChebNet architecture, as well as motivate this specific choice for the architecture we will subsequently introduce.

3.1 Calculus on Graphs

Let $\mathcal{G} = (V, E)$ be an undirected graph with $|V| = n$ vertices and $|E| = m$ edges. We denote by $\mathcal{H}_V$ the Hilbert space of square-integrable *node signals* $f : V \to \mathbb{R}$, with corresponding scalar product $\langle f, g\rangle := \sum_{i=1}^{n} f(i)g(i)$, for i running over the nodes in V. Likewise, $\mathcal{H}_E$ will denote the Hilbert space of square-integrable *edge signals* $G : E \to \mathbb{R}$. To define signed edge differences, we fix an arbitrary orientation of each undirected edge. This orientation is only a bookkeeping device and does not affect the Laplacian obtained below. The corresponding scalar product in $\mathcal{H}_E$ is given by $\langle\!\langle F, G\rangle\!\rangle := \sum_{e=1}^{m} F(e)G(e)$, for e running over the edges in E. We will reserve the use of capital letters for such edge signals.

For an oriented edge $e = (i, j)$, the *gradient operator on* $\mathcal{G}$ is denoted $\nabla^{\mathcal{G}} : \mathcal{H}_V \to \mathcal{H}_E$ and acts on a node signal $f \in \mathcal{H}_V$ as

$$\nabla^{\mathcal{G}}_{ij} f := f(j) - f(i) \tag{1}$$

for two connected nodes $i, j \in V$. The adjoint of $\nabla^{\mathcal{G}}$ with respect to $\langle\cdot,\cdot\rangle$ as above is given by $\nabla^{\mathcal{G}\star} : \mathcal{H}_E \to \mathcal{H}_V$ acting on $F \in \mathcal{H}_E$ as

$$(\nabla^{\mathcal{G}^*} F)(i) = \sum_{e=(j,i)} F(e) - \sum_{e=(i,j)} F(e), \tag{2}$$

where the first sum is over edges oriented toward i and the second over edges oriented away from i.

Proposition 1. *For any $f \in \mathcal{H}_V$ and $F \in \mathcal{H}_E$, we have $\langle\langle \nabla^{\mathcal{G}} f, F \rangle\rangle = \langle f, \nabla^{\mathcal{G}^*} F \rangle$.*

Proposition 2. *The positive graph Laplacian induced by the gradient is $L := -\mathrm{div}^{\mathcal{G}} \nabla^{\mathcal{G}}$, where the divergence operator is given by $\mathrm{div}^{\mathcal{G}} := -\nabla^{\mathcal{G}^*}$. We immediately notice that this is analogous to the Euclidean construction of the Laplace operator $\Delta := \mathrm{div}\nabla$. Moreover, we have*

$$L = D - A, \tag{3}$$

where A is the adjacency matrix of $\mathcal{G}$ and D is the degree matrix with entries $D_{ii} = \sum_{j=1}^{n} A_{ij}$. This matches the well-known definition of the combinatorial graph Laplacian *(Chung, 1997).*

L encodes the connectivity structure of the graph, is symmetric, positive semidefinite, and admits an eigenvalue decomposition of the form $L = U\Lambda U^\top$, where $\Lambda = \mathrm{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ is the diagonal matrix of ordered eigenvalues $0 \le \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1}$ and U is an orthogonal matrix of eigenvectors.

3.2 Weighted Diffusions Steer & Propagate Information

The diffusion equation models the evolution of a quantity $u : \mathbb{R}^d \times \mathbb{R}_{\ge 0} \to \mathbb{R}$ that spreads out depending on a diffusion term. Starting from the conservative continuity equation

$$\partial_t u(x,t) = -\nabla \cdot j(x,t), \tag{4}$$

where $j(x,t) = -\mu(x)\nabla u(x,t)$ and $\mu : \mathbb{R}^d \to \mathbb{R}_{>0}$ is the spatially-dependent diffusivity, we get the following evolution

$$\partial_t u(x,t) = \mu(x)\Delta u(x,t) + \nabla\mu(x) \cdot \nabla u(x,t). \tag{5}$$

When $\mu \equiv 1$, Eq. (5) reduces to classical heat dissipation $\partial_t u = \Delta u$.When μ varies in space, however, the right-hand side of (5) develops a first-order term and therefore has the form of a drift or advection contribution. In this sense, an apparently purely diffusive equation can induce drift-like dynamics once diffusion takes place in an inhomogeneous medium.

This observation is the basis of our approach: instead of prescribing an external drift, we *learn the geometry that induces it*. A positive node-wise weight function μ biases diffusion such that drift-like transport emerges from the operator itself. Learned from data, μ becomes a task-dependent geometric bias that steers information flow through the graph.

3.3 The ChebNet Architecture

The previous section suggests a route to task-adaptive propagation: rather than solving an explicit advection-diffusion PDE or introducing attention-like transport mechanisms, one may seek to encode the transport bias directly into the diffusion operator $\Delta \mapsto \mu\Delta$. On graphs, the natural diffusion operator is the Laplacian L from Equation (3). This makes spectral GNNs in particular a natural setting for our construction, since their propagation rules are built directly from L and its spectrum. Any principled modification of the Laplacian immediately induces a corresponding modification of the propagation geometry, while leaving the spectral architecture essentially unchanged. This makes ChebNet an ideal backbone for our construction: once the learned weight function μ is used to bias the graph Laplacian $L \mapsto L_\mu$, the resulting geometry can be inserted directly into the Chebyshev filtering pipeline. More details on the ChebNet architecture can be found in the Appendix D.

4 From Weighted Diffusion to a Node-Weighted Graph Laplacian

We now show how to construct biased diffusions on graphs, as done in Section 3.2.

4.1 Weighted Graph Laplacian

Indeed, introducing an external drift term of the form $v(x) \cdot \nabla u(x,t)$, where $v : \mathbb{R}^d \to \mathbb{R}^d$ is a vector field, into Eq. (5) is not well-defined on graph domains, unlike the continuous setting

Desbrun et al. (2005). Instead, we turn to Dirichlet forms, which admit both $\mathbb{R}^d$ and graph analogs. In finite dimensions, once an inner product $\langle \cdot, \cdot \rangle$ is fixed, a Dirichlet form $\mathcal{D}$ is in one-to-one correspondence (Fukushima et al., 2011) with a self-adjoint positive semidefinite Laplacian L through the representation $\mathcal{D}(f, g) = \langle f, Lg \rangle$, where $f, g \in L^2(\mathbb{R}^d; \mathbb{R})$ in the continuous case, and $f, g \in \mathcal{H}_V$ in the graph case.

Continuous Case. Let $\Omega \subset \mathbb{R}^d$ be a bounded open domain, and let $f, g : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ be smooth functions vanishing on $\partial\Omega$. For a positive weight function $\mu : \Omega \to \mathbb{R}_{>0}$, the Dirichlet form given by

$$\mathcal{D}_\mu(f, g) := \frac{1}{2} \int_\Omega \mu(x) \nabla f(x) \cdot \nabla g(x) dx = \langle f, \tilde{\Delta}_\mu g \rangle_{L^2(dx)}, \tag{6}$$

is in one-to-one correspondence to the second-order operator

$$\tilde{\Delta}_\mu f = -\frac{1}{2} \mu \Delta f - \frac{1}{2} \nabla \mu \cdot \nabla f, \tag{7}$$

when considering the usual L^2 inner product. This particular choice of inner product allows us to recover the emerging first-order drift term, previously derived in Eq. (5). We now adapt to graphs.

Graph Case. Analogously to the continuous case above, the Dirichlet form on a graph $\mathcal{G}$ with positive node-wise weight function $\mu : V \to \mathbb{R}_{>0}$ is given by

$$\mathcal{D}_\mu^{\mathcal{G}}(f, g) = \frac{1}{2} \sum_i \mu_i \sum_{j \sim i} \nabla_{ij}^{\mathcal{G}} f \, \nabla_{ij}^{\mathcal{G}} g, \tag{8}$$

where $\nabla^{\mathcal{G}}$ is defined as in Eq. (1), and f and g graph signals. In the following result, we derive the corresponding graph operator L_μ in closed form, associated to the edge signal inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ from Section 3. The proof is detailed in Appendix C.3.

Theorem 1 (Closed form of L_μ)**.** *Let $\mu : V \to \mathbb{R}_{>0}$ and let $\mathcal{D}_\mu^{\mathcal{G}}$ be the weighted graph Dirichlet form defined in* (8)*. Then there exists a unique symmetric positive semidefinite matrix L_μ such that $\mathcal{D}_\mu^{\mathcal{G}}(f, g) = f^\top L_\mu g$. **Moreover, L_μ is the weighted graph Laplacian***

$$L_\mu = D_\mu - A_\mu, \tag{9}$$

where $A_\mu = M_\mu \odot A$, $(M_\mu)_{ij} = \frac{\mu_i + \mu_j}{2}$, and $D_\mu = \mathrm{diag}(A_\mu \mathbf{1})$.

Theorem 1 is the weighted analogue of Eq. (3): the standard Laplacian $L = D - A$ is recovered as the special case $A_\mu = A$. The theorem shows that the edge weights are induced by the node-wise weight function μ: each edge (i, j) receives $\frac{1}{2}(\mu_i + \mu_j)$ from its two endpoint values. In particular, the original graph topology is preserved, since $A_{ij} = 0$ implies $(A_\mu)_{ij} = 0$. The weight function therefore changes only the ease of transport along existing edges. The next result makes the analogy with the continuous identity from Eq. (7) explicit. See Appendix C.4 for a detailed proof.

Theorem 2 (Drift-diffusion decomposition)**.** *Using the graph calculus from Section 3.1, the action of L_μ can be written as*

$$(L_\mu f)_i = \mu_i (L f)_i - \frac{1}{2} \sum_{j \sim i} \nabla_{ij}^G \mu \, \nabla_{ij}^G f, \tag{10}$$

where L is the graph Laplacian as in Eq. (3)*.*

The first term on the right-hand side of Eq. (10) is the usual graph diffusion scaled by the local value μ_i. The second term weights each edge difference of the signal by the corresponding edge difference of μ, so it appears only where the learned density and the signal vary. It is therefore a discrete first-order, drift-like correction induced by the variation of μ. In this sense, it biases diffusion toward selected regions of the graph without introducing an explicit vector field and without modifying the adjacency. See Appendices A and E for details and further visualizations of the learned density μ.

4.2 Spectral properties and finite-time diffusion

We now study how the geometry induced by μ affects the two propagation limitations targeted in this work. We show that reshaping the spectrum changes the finite-time smoothing dynamics, enabling it

to delay oversmoothing, and we show that the same node weighting can reduce effective resistance across graph bottlenecks, a quantity linked to oversquashing.

Theorem 3 (Spectral comparison). *Let $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1}$ be the eigenvalues of L, and let λ_k^μ denote the eigenvalues of L_μ, ordered similarly. For each $k \in \{0, \dots, n-1\}$, there exist positive constants $c_k^\mu \le C_k^\mu$, defined in Appendix C.5, such that*

$$\lambda_k \|\mu\|_1 c_k^\mu \;\le\; \lambda_k^\mu \;\le\; \lambda_k \|\mu\|_1 C_k^\mu. \tag{11}$$

The effect of μ on the spectrum is therefore mode dependent. As established in Appendix C.7, the constants in Theorem 3 further control low-frequency eigenspace stability. This guarantees that internal mode-mixing does not destabilize eigenspaces, preserving robustness even on highly symmetric graphs with repeated eigenvalues. We now examine these modal structures.

Spectral Control of Finite-Time Smoothing. The learnable weight function μ does not prevent asymptotic smoothing. It reshapes the spectrum so that, over the finite propagation horizon used by Chebyshev filters, task-relevant low-frequency modes decay more slowly relative to nuisance/high-frequency modes. This creates a transient window in which informative components persist before eventual collapse. We quantify this transient regime through finite-time modal analysis and detail how Theorem 3 actually provides crucial insights. For the sake of argumentation, let us explain this with the heat diffusion dynamics as a model of linear propagation: $x(t) = e^{-tL_\mu}x(0) = \sum_{k=0}^{n-1} e^{-t\lambda_k^\mu}\langle u_k^\mu, x(0)\rangle u_k^\mu$. On an Stochastic-Black-Model-like (SBM) graph, we identify the Fiedler vector as the informative mode carrying the graph partition. As noted in Appendix C.7 (Proposition 3), when the perturbation is small compared with the spectral gap, u_k^μ remains close to u_k. We can therefore compare the decay of the same spectral mode under L and L_μ. The decay of mode k relative to the Fiedler mode is given by the ratio of their modal velocities: $\frac{e^{-t\lambda_k^\mu}}{e^{-t\lambda_1^\mu}} = e^{-t\lambda_1^\mu(\rho_k^\mu - 1)} = e^{-\tau(\rho_k^\mu - 1)}$ with $\rho_k^\mu = \frac{\lambda_k^\mu}{\lambda_1^\mu}$ and $t\lambda_1^\mu = \tau$ the new reference time scale. For a fixed global diffusion scale, a learned potential can reduce this ratio for task-relevant modes, extending the time interval during which those modes remain comparable to the bottleneck or community nodes. This is what we mean by "delayed diffusion" in the paper. We therefore would like $\rho_k^\mu < \rho_k$ for task-relevant modes. For a visual example, we ran the following experiment. We generated a 400 nodes SBM graph with two balanced communities and computed its combinatorial Laplacian. Intra-community edge probabilty was set at $p_{\text{intra}} = 0.1$ and inter-community edge probability at $p_{\text{inter}} = 0.005$. Further, we manually created a weight function μ with constant values $C_1, C_2 > 0$ on respective community nodes and computed the corresponding weighted Laplacian L_μ. Figure 1 displays the relative modal velocities ρ_k and ρ_k^μ. First, low-frequency modes (small k) exhibit lower relative velocities for L_μ. These modes include the bottleneck-aligned components responsible for large-scale structure. They are preserved for a longer diffusion time/number of layers. Conversely, high-frequency modes (large k), which are essentially intra community, uninformative modes have significantly larger relative velocities for L_μ than in the unweighted case. They are therefore erased faster relative to the Fiedler mode. A more detailed analysis of the example linked to Theorem 3 is given in Appendix F.

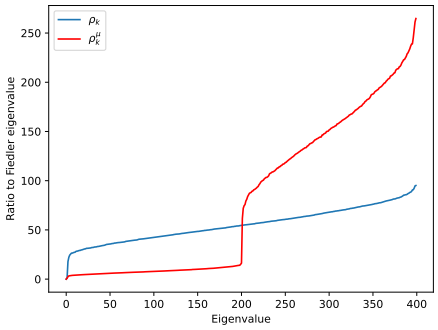


Figure 1: Relative modal velocities ρ_k (blue) and ρ_k^μ (red) for the synthetic SBM experiment.

Spectral Control under Nonlinearities. We now show that the spectral control established in Theorem 3 persists in the presence of nonlinearities. A simplified version of the result is reported below. A precise formulation and proof are available in Appendix C.8.

Theorem 4. *Let $\mathbf{z}$ be a pre-activation state on the graph, $\mathbf{h}$ be an infinitesimal perturbation localized around the k-th mode, and σ be a Lipschitz activation function such that $0 < m \le$*

$\sigma'(x) \leq M$. Let $\mathbf{h}^+ = diag(\sigma'(\mathbf{z}))\mathbf{h}$ define the linearized post-activation perturbation. If the activation function satisfies $\mathcal{D}^{\mathcal{G}}_{\mu}(\sigma'(\mathbf{z})) \leq \varepsilon$ across the edges, then the μ-weighted Dirichlet energy of the linearized post-activation perturbation $\mathbf{h}^+$ satisfies:

$$m^2 \frac{c_k^{\mu}}{C_k^{\mu}} \mathcal{D}^{\mathcal{G}}_{\mu}(\mathbf{h}) - \mathcal{O}(\varepsilon) \;\leq\; \mathcal{D}^{\mathcal{G}}_{\mu}(\mathbf{h}^+) \;\leq\; M^2 \frac{C_k^{\mu}}{c_k^{\mu}} \mathcal{D}^{\mathcal{G}}_{\mu}(\mathbf{h}) + \mathcal{O}(\varepsilon). \tag{12}$$

The bound (12) demonstrates that the presence of an activation does not induce unconstrained spectral leakage. In the $\epsilon \to 0$ limit, Eq. (12) acts as a bounded filter confining the k-th mode to a corridor set by the activation's slope range $[m, M]$ and the same condition number $\delta_k^{\mu} = C_k^{\mu}/c_k^{\mu}$ through which μ reshapes propagation (Theorem 3). Geometry and nonlinearity are therefore governed by one consistent quantity. Roughness of the activation $\sigma'(z)$ controls the tightness of Eq. (12). As node features typically smoothen with depth, this suggests that the corridor tightens in later layers, reinforcing rather than eroding μ's mode-selective propagation mechanism. We leave a multi-layer analysis to future work.

Node Weighting and Oversquashing. Beyond oversmoothing, μ also affects bottlenecks associated with oversquashing. Black et al. (2023) relate lower effective resistance between node pairs to less oversquashing in MPNNs. Figure 2 shows that there exists a choice of μ that lowers the resistance distance between nodes belonging to opposite cliques, indicating that the learned geometry can alleviate bottlenecks without modifying the graph topology.

5 A NOVEL CLASS OF SPECTRAL GNNS

The proposed μ-Laplacian defines a learnable diffusion geometry that can be integrated into graph neural architectures relying on propagation operators, including diffusion-based and message-passing models. Even though it is not tied to a spectral parameterization, we chose to instantiate it with a simple ChebNet architecture due to its good performances and limited computational complexity.

Learnable Weight Function. As discussed in Section 4.2, the weight function μ allows control of the spectral properties of L_{μ}, but it usually remains difficult to design it by hand in practice. We therefore propose a two-step strategy. First, we make a *neural parameterization of* μ. We choose a simple but sufficiently expressive GCN: $\mu = \text{GCN}(A, X, \Theta)$, where A is the adjacency matrix of $\mathcal{G}$, X is a feature tensor, and Θ consists of learnable weights.

μ-ChebNet Filter. Once μ is calculated, the second step is to construct L_{μ} using Eq. (10), and use it directly into a downstream spectral ChebNet. This leads to a new family of architectures, with varying parameterization of μ and downstream ChebNet that can be learned end-to-end to solve a given task. At training time, the weight function μ will be adapted so the downstream μ-ChebNet solves the task and will thus depend on the task and distribution of training samples. As we shall see below, inspecting μ as a graph signal reveals much about the strategy employed by the network.

6 EXPERIMENTS

We investigate the learned Laplacian in three stages. We first examine what kind of geometry is learned with μ, independently of task performance. We then use a synthetic barbell task to test whether the resulting adaptation translates into improved performance over a fixed-Laplacian ChebNet across different architectural depths. Finally, we evaluate the model on a real-world benchmark.

Resistance Distance examination with Learned Weight Function. Having previously shown that a heuristically chosen μ can reduce the intercommunity resistance distance (Figure 2), we now examine whether this behavior also emerges from the learned node weights. Figure 3 shows that the learned μ, here in barbell node prediction task, indeed recovers the effect observed with the hand-crafted μ. This indicates that the learned geometry can alleviate the structural bottleneck.

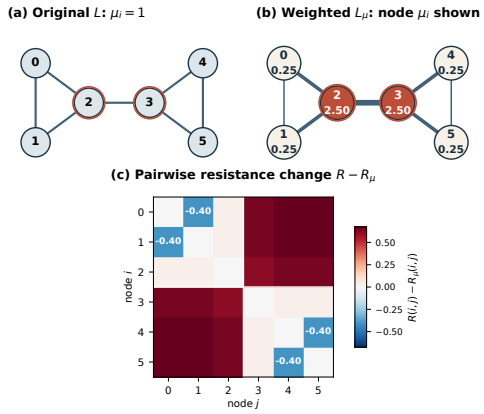


Figure 2: Heuristically defined μ in a barbell graph shows existence of μ that reduces the cross-clique resistance. Red indicates lower resistance under L_μ; blue indicates higher resistance.

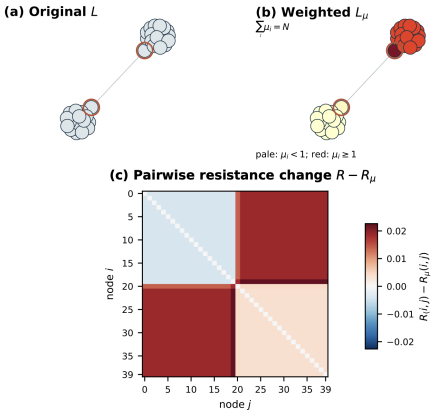


Figure 3: The learned μ in representative node prediction task reduces the cross-clique resistance. Red indicates lower resistance under L_μ; blue indicates higher resistance.

Oversquashing assessment on Barbell Graphs. The synthetic *Barbell dataset* (Bamberger et al., 2024) probes oversquashing by requiring nodes in each clique to predict features from the opposite clique, while all cross-clique information must pass through a single bridge edge. Thus, even when the message-passing depth is sufficient for reachability, standard MPNNs must compress many source signals through a narrow graph bottleneck, exposing their inability to transmit long-range information without severe distortion.

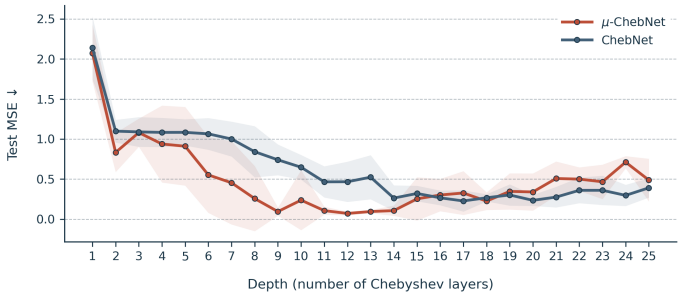


Figure 4: Effects of μ and network depth on the barbell benchmark. Curves and shaded regions report mean and standard deviation over 5 replicates. μ-ChebNet achieves lower error and reaches its optimal regime at lower depth than ChebNet. Increasing the depth of the vanilla model does not recover comparable performance, suggesting that adapting the propagation geometry is more effective than increasing the number of layers.

Results. We observe in Figure 4 that μ-ChebNet opens a finite-time window—which, in the discrete setting, translates into a "layer window"—during which it outperforms ChebNet. Theorem 3 shows that the learned Laplacian can alter the relative decay rates of its eigenmodes, allowing some of them to persist longer relative to others. The improved performance observed here suggests that learning μ favors modes that are useful for the task. As expected, this advantage persists only over a finite window, since the underlying dynamics remain diffusive and eventually smooth these modes as well. Crucially, μ-ChebNet reaches its best performance within this shallow regime, whereas vanilla ChebNet does not attain comparable performance even at greater depths. This suggests that adapting the propagation geometry can reduce the number of propagation steps required to solve the long-range task.

Real-World Evaluation on Open-Graph Benchmark: Proteins. The `ogbn-proteins` dataset (Hu et al., 2020) is a biologically meaningful benchmark, large-scale graph with multi-label protein function prediction tasks on a protein-protein interaction network.

Results. We observe in Table 1 that μ-ChebNet reaches a score of 80.61%, outperforming the MPNN baselines (Kipf, 2016; Veličković et al., 2018; Hamilton et al., 2017; Xu et al., 2019),

and remaining competitive with both Stable-ChebNet (Hariri et al., 2026) and Graph Transformer architectures (Bo et al., 2023; Wu et al., 2022; 2023b). Although SPEXPHORMER obtains comparable performance, our model relies only on ordinary sparse graph operations. Computationally, μ-ChebNet incurs the cost of a shallow GCN used to learn μ, an $O(m)$ construction of L_μ, and a downstream Chebyshev filter whose propagation cost scales as $O(Km)$, up to feature-channel mixing. Hence, on sparse graphs with fixed K and bounded width, the overall complexity is linear in the number of nodes. By contrast, SPEXPHORMER reduces the final attention layer to $O(nd^2 + ndc)$ by selecting c attention edges per node, but requires an additional attention-estimation stage. Thus, while both approaches can scale linearly on sparse graphs, our approach obtains task-adaptive propagation through a learned sparse Laplacian and avoids a two-stage attention-sparsification pipeline. Moreover, Figure 5 features the same finite-depth advantage already observed on synthetic tasks.

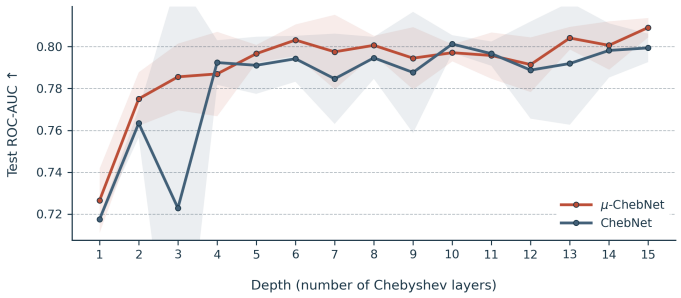


Figure 5: Effects of μ and network depth on the `ogbn-proteins`. Curves and shaded regions report mean and standard deviation over 5 replicates.

Discussion and Future Work. Across the experiments, the empirical results support the theoretical demonstrations and analysis: μ-ChebNet is most effective on long-range and bottlenecked tasks, where learning L_μ improves over fixed-Laplacian baselines. It facilitates information propagation even with limited model depth. On real-world benchmark, it rivals larger architectures such as graph transformers, while remaining a lightweight end-to-end model. We believe that further important future research lies in deriving tighter eigenvalue bounds for L_μ, extending the framework over other downstream spectral architectures beyond ChebNet-style filters, and ablating over alternative parameterizations of μ beyond the GCN considered in this work.

Table 1: Competitive performance on `ogbn-proteins`; baselines from Hariri et al. (2025).

Model	**ROC-AUC ↑**
MLP	72.04 ± 0.48
GCN	72.51 ± 0.35
ChebNet	77.55 ± 0.43
Stable-ChebNet	79.55 ± 0.34
SGC	70.31 ± 0.23
GCN-NSAMPLER	73.51 ± 1.31
GAT-NSAMPLER	74.63 ± 1.24
SIGN	71.24 ± 0.46
NodeFormer	77.45 ± 1.15
SGFormer	79.53 ± 0.38
SPEXPHORMER	$\mathbf{80.65 \pm 0.07}$
μ-ChebNet (ours)	$\mathbf{80.61 \pm 0.01}$

7 Conclusion

In this work, we achieve guided diffusion dynamics on graphs by learning a data-driven weighting μ of the graph nodes, resulting in a task-adaptive graph Laplacian $L_\mu = D_\mu - A_\mu$ (Theorem 1). This produces an end-to-end rewiring-like effect, without changing the graph topology. We prove that this learnable weight function μ allows task-dependent control over the graph's spectrum by changing the relative decay rates of graph Fourier modes (Theorem 3). We further extend our analysis to the nonlinear setting, showing that this spectral control persists beyond linear diffusion and can therefore influence information propagation in neural networks. Moreover, we introduce μ-ChebNet, an upgraded ChebNet architecture incorporating the modulable graph Laplacian L_μ. We experimentally observe that μ-ChebNet drastically outperforms current ChebNet architectures on signal propagation through narrow bottlenecks (Barbell Graph), reaching at shallow depths performance that vanilla ChebNet does not recover even with additional layers. On real-world datasets, μ-ChebNet also performs comparably to state-of-the-art architectures. Our work opens doors to further exploring the role of task-adaptive geometry in machine learning, offering better interpretability at a lower cost.

AI USE STATEMENT

Generative AI tools were used to improve the grammar and clarity of the manuscript, to assist in checking algebraic derivations and mathematical arguments, and to improve the visual consistency of figures. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria M Oliver. DiffWire: Inductive Graph Rewiring via the Lovász Bound. In Bastian Rieck and Razvan Pascanu (eds.), *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pp. 15:1–15:27. PMLR, 09–12 Dec 2022.

Álvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael Bronstein, and Pierre Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in gnns: Bridging recurrent and graph learning. *arXiv preprint arXiv:2502.10818*, 2025.

Hugo Attali, Davide Buscaldi, and Nathalie Pernelle. Rewiring techniques to mitigate oversquashing and oversmoothing in gnns: A survey. *arXiv preprint arXiv:2411.17429*, 2024.

Jacob Bamberger, Federico Barbero, Xiaowen Dong, and Michael Bronstein. Bundle neural networks for message diffusion on graphs. *arXiv preprint arXiv:2405.15540*, 2024.

Federico Barbero, Ameya Velingker, Amin Saberi, Michael M. Bronstein, and Francesco Di Giovanni. Locality-aware graph rewiring in GNNs. In *The Twelfth International Conference on Learning Representations*, 2024.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *International Conference on Machine Learning*, pp. 2528–2547. PMLR, 2023.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. *arXiv preprint arXiv:2303.01028*, 2023.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Benjamin Chamberlain, James Rowbottom, Davide Eynard, Francesco Di Giovanni, Xiaowen Dong, and Michael Bronstein. Beltrami flow and neural diffusion on graphs. *Advances in Neural Information Processing Systems*, 34:1594–1609, 2021a.

Benjamin Paul Chamberlain, James Rowbottom, Maria Gorinova, Stefan Webb, Emanuele Rossi, and Michael M Bronstein. GRAND: Graph neural diffusion. In *International Conference on Machine Learning (ICML)*, pp. 1407–1418. PMLR, 2021b.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. *arXiv preprint arXiv:2206.04910*, 2022.

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

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

Mathieu Desbrun, Anil N Hirani, Melvin Leok, and Jerrold E Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.

Stefano Fiorini, Stefano Coniglio, Michele Ciavotta, and Enza Messina. Sigmanet: One laplacian to rule them all. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7568–7576, 2023.

Masatoshi Fukushima, Yoichi Oshima, and Masayoshi Takeda. *Dirichlet forms and symmetric Markov processes*, volume 19. Walter de Gruyter, 2011.

Benjamin Girault, Antonio Ortega, and Shrikanth S Narayanan. Irregularity-aware graph fourier transforms. *IEEE Transactions on Signal Processing*, 66(21):5746–5761, 2018.

Chen Gong, Tongliang Liu, Dacheng Tao, Keren Fu, Enmei Tu, and Jie Yang. Deformed graph laplacian for semisupervised learning. *IEEE transactions on neural networks and learning systems*, 26(10):2261–2274, 2015.

Alessio Gravina, Davide Bacciu, and Claudio Gallicchio. Anti-Symmetric DGN: a stable architecture for Deep Graph Networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Benjamin Gutteridge, Xiaowen Dong, Michael M Bronstein, and Francesco Di Giovanni. Drew: Dynamically rewired message passing with delay. In *International Conference on Machine Learning*, pp. 12252–12267. PMLR, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Ali Hariri, Álvaro Arroyo, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, Davide Bacciu, Kamyar Azizzadenesheli, Xiaowen Dong, and Pierre Vandergheynst. Return of ChebNet: Understanding and Improving an Overlooked GNN on Long Range Tasks, October 2025. arXiv:2506.07624 [cs].

Ali Hariri, Alvaro Arroyo, Alessio Gravina, Moshe Eliasof, Carola-Bibiane Schönlieb, Davide Bacciu, Xiaowen Dong, Kamyar Azizzadenesheli, and Pierre Vandergheynst. Return of chebnet: Understanding and improving an overlooked gnn on long range tasks. *Advances in Neural Information Processing Systems*, 38:136166–136196, 2026.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20. Curran Associates Inc., 2020. ISBN 9781713829546.

Yufei Jin and Xingquan Zhu. Oversmoothing alleviation in graph neural networks: a survey and unified view: Y. jin, x. zhu. *Knowledge and Information Systems*, 67(12):11259–11285, 2025.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. arXiv:1609.02907 [cs].

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Ron Levie, Wei Huang, Lorenzo Bucci, Michael Bronstein, and Gitta Kutyniok. Transferability of spectral graph convolutional neural networks. *J. Mach. Learn. Res.*, 22(1), January 2021. ISSN 1532-4435.

Yu Li, Meng Qu, Jian Tang, and Yi Chang. Signed laplacian graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4444–4452, 2023.

Sohir Maskey, Raffaele Paolino, Aras Bacho, and Gitta Kutyniok. A fractional graph laplacian approach to oversmoothing. *Advances in Neural Information Processing Systems*, 36:13022–13063, 2023.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A Survey on Oversmoothing in Graph Neural Networks. *arXiv preprint arXiv:2303.10993*, 2023.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Zhiqi Shao, Dai Shi, Andi Han, Yi Guo, Qibin Zhao, and Junbin Gao. Unifying over-smoothing and over-squashing in graph neural networks: A physics informed approach and beyond. *arXiv preprint arXiv:2309.02769*, 2023.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pp. 31613–31632. PMLR, 2023.

Hamed Shirzad, Honghao Lin, Balaji Venkatachalam, Ameya Velingker, David P Woodruff, and Danica J Sutherland. Even sparser graph transformers. *Advances in Neural Information Processing Systems*, 37:71277–71305, 2024.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401, 2022.

Qitian Wu, Chenxiao Yang, Kaipeng Zeng, Fan Nie, Michael Bronstein, and Junchi Yan. Advective diffusion transformers for topological generalization in graph learning. *arXiv preprint arXiv:2310.06417*, 2023a.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023b.

Qitian Wu, David Wipf, and Junchi Yan. Transformers from diffusion: A unified framework for neural message passing. *Journal of Machine Learning Research*, 26(129):1–47, 2025a.

Qitian Wu, Chenxiao Yang, Kaipeng Zeng, and Michael M Bronstein. Supercharging graph transformers with advective diffusion. In *Forty-second International Conference on Machine Learning*, 2025b.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew Hirn. Magnet: A neural network for directed graphs. *Advances in neural information processing systems*, 34: 27003–27015, 2021.

A ILLUSTRATING ENHANCED EXPLAINABILITY

We provide further evidence that the learned weight function μ helps revealing the strategy used by the network to solve tasks. We consider a dataset of ring graphs and a task that consists in predicting the correct class (among 10) of a query node. This class is encoded as a Dirac in the features of a distant answer node. The query and answer nodes are symmetrically positioned so that there are two long paths of equal lengths connecting them. The features along the first path are all-zero, but the features along the second are i.i.d Gaussian noise. Clearly, agglomerating information along that second path results in accumulating noise and therefore poor decision making. As can be seen on Figure 6, μ assigns higher values to vertices on the most efficient path from the answer node to the query node — effectively steering signal propagation along meaningful routes, without altering the original graph topology. Specifically, μ successfully suppresses one of the two paths, concentrating signal flow along a single route. Beyond improving predictive performance, the learned weight functions offer transparent, interpretable insights into how the model routes information across complex structures.

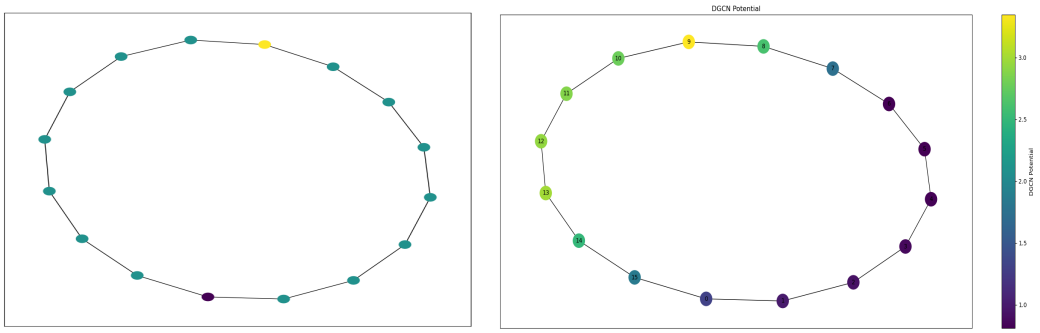


Figure 6: Ring graph configuration. (Left) The query node is in purple and the answer node in yellow. Nodes on the rightward path contain noisy features. (Right) The resulting learned weight function has broken the symmetry of the configuration, routing information on the most informative path.

B ANALYSIS COMMUNITY GRAPH

Avoiding Repeated Eigenvalues We consider the graph formed by two triangles connected by a single bridge edge. This graph is a minimal model of a bottlenecked community structure: each triangle is internally dense, while all inter-community communication must pass through one edge. For the standard combinatorial Laplacian, the spectrum is

$$\{0,\ 0.438,\ 3,\ 3,\ 3,\ 4.562\}.$$

The repeated eigenvalue at 3 is caused by the symmetry of the two communities. Under heat diffusion, all modes associated with this repeated eigenvalue decay at the same rate. The Fiedler eigenvector, associated with the first non-zero eigenvalue, is aligned with the partition between the two triangles and has a large edge difference on the bridge. We then choose a piecewise constant density μ, with $\mu = 1.5$ on one community and $\mu = 0.5$ on the other, to yield $\sum_i \mu_i = 6 = \sum_{i=1}^{6} 1$ as for the unweighted case. The spectrum of L_μ becomes

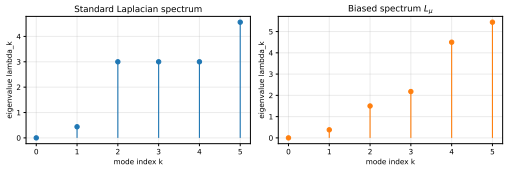
$$\{0,\ 0.38,\ 1.5,\ 2.18,\ 4.5,\ 5.44\}.$$

The weighted operator removes the eigenvalue degeneracy and modifies the eigenvectors. Modes whose variations lie mostly in the high-μ region become more costly and are shifted toward larger eigenvalues, while modes varying mostly in the low-μ region remain comparatively slower. Normalizing by the Fiedler eigenvalue gives the relative decay rates

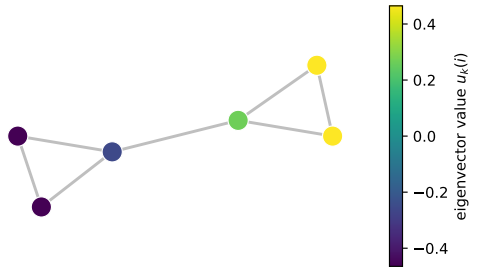
$$L : \{0,\ 1,\ 6.84,\ 6.84,\ 6.84,\ 10.40\},$$

whereas

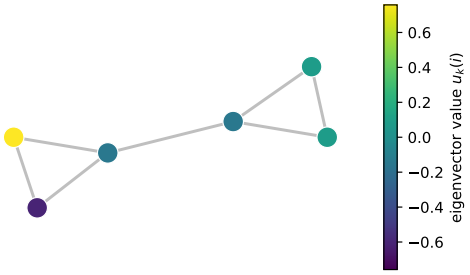
$$L_\mu : \{0,\ 1,\ 3.17,\ 5.70,\ 15.87,\ 18.69\}.$$



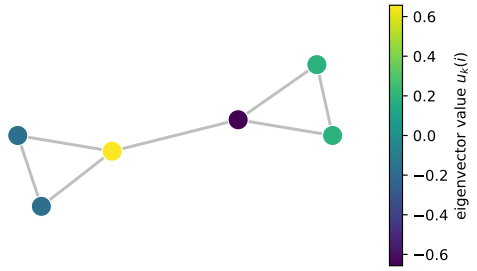
(a) Eigenvalue spectra of the standard Laplacian L and biased Laplacian L_μ. The repeated eigenvalue in L is split under L_μ, showing symmetry breaking.



(b) Fiedler mode (λ_1): aligned with the community partition.



(c) Representative middle-band mode (degenerate band of L).



(d) High-frequency mode.

Figure 7: Spectral analysis on the simple community graph.

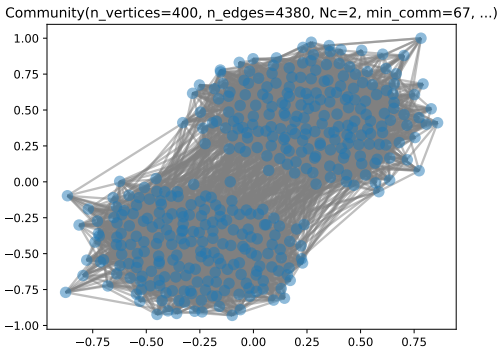


Figure 8: 2-community Graph

This shows that L_μ does not simply accelerate or slow all modes uniformly. Instead, it changes the relative timing of spectral decay: some components become slower relative to the partition mode, while others are damped much faster, as shown in Figure 7. This example highlights the transient nature of the mechanism. Both L and L_μ eventually drive heat diffusion toward the constant eigenspace, so neither operator prevents collapse at infinite time. The difference lies in the finite-time regime relevant for GNNs: L_μ changes which modes remain available before collapse occurs. In this sense, the learned density steers propagation by re-timing spectral decay rather than by inducing faster-than-diffusive transport.

Oversquashing Mitigation via Learned Diffusion Geometry To illustrate how the weighted Laplacian L_μ helps mitigate oversquashing, we perform the following experiment: We generate a 2-community graph C_1 and C_2 with 400 nodes and a strong bottleneck as shown in Figure 8. We choose $\mu = c_1$ on community C_1 and $\mu = c_2$ on community C_2, where c_1, c_2 are two positive

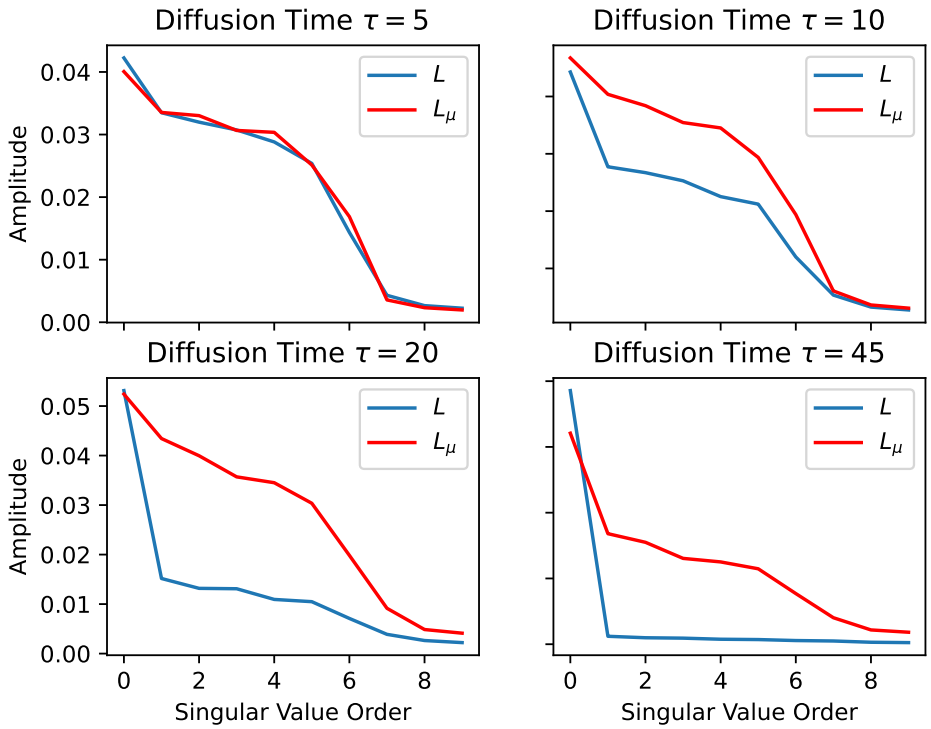


Figure 9: Singular values of the propagated feature matrix restricted to the target cluster. While standard diffusion rapidly collapses all but one direction, the learned weighted diffusion geometry opens a transient regime in which multiple independent directions survive long enough to traverse the bottleneck.

constants. Further, we initiate a feature matrix X_0 consisting of k orthogonal vectors localized on one community. We then propagate using the heat kernels $e^{\tau L}$ and $e^{\tau L_\mu}$ for the weighted Laplacian applied to X_0. As propagation time τ grows, the signals will transit through the bottleneck to reach the second community. Since the bottleneck is strong, it will limit the flow of information leading to oversquashing. To measure this phenomenon, we track the singular values of the signal restricted to the second community. Under standard diffusion, the effective rank collapses rapidly, indicating severe information compression. In contrast, the weighted Laplacian opens a clear transient regime in which several singular values remain significant, allowing multiple independent directions to traverse the bottleneck before collapse. This demonstrates that the biased diffusion geometry alleviates oversquashing by reallocating spectral mass away from local oscillations and toward bottleneck-aligned transport, as illustrated in Figure 9.

C Proofs

C.1 Proof of Proposition 1

Taking the definitions of ∇^G and $\nabla^{G^\star}$ from Eqs. (1) and (2), we obtain

$$\langle\!\langle \nabla^{\mathcal{G}} f, F \rangle\!\rangle = \sum_{e=(i,j)\in E} (f(j) - f(i))F(e) = \sum_{i\in V} f(i) \left(\sum_{e=(j,i)} F(e) - \sum_{e=(i,j)} F(e) \right) = \langle f, \nabla^{\mathcal{G}^\star} F \rangle. \tag{13}$$

proving the claim.

□

C.2 Proof of Proposition 2

For any node i,

$$(\nabla^{\mathcal{G}^*}\nabla^{\mathcal{G}} f)(i) = \sum_{e=(j,i)} (\nabla^{\mathcal{G}} f)(e) - \sum_{e=(i,j)} (\nabla^{\mathcal{G}} f)(e).$$

If $e = (j,i)$, then $(\nabla^{\mathcal{G}} f)(e) = f(i) - f(j)$. If $e = (i,j)$, then $(\nabla^{\mathcal{G}} f)(e) = f(j) - f(i)$, so the second term contributes $f(i) - f(j)$. Hence

$$(\nabla^{\mathcal{G}^*}\nabla^{\mathcal{G}} f)(i) = \sum_{j\sim i}(f(i) - f(j)) = (D - A)f(i).$$

Therefore $\nabla^{\mathcal{G}^*}\nabla^{\mathcal{G}} = D - A$. The divergence identity follows from $\mathrm{div}^{\mathcal{G}} := -\nabla^{\mathcal{G}^*}$.

□

C.3 Proof of Theorem 1

We first consider a non-negative node-wise weight $\mu : V \to \mathbb{R}^+$ and the corresponding Dirichlet bilinear form: $\mathcal{D}^G_\mu(f,g) = \frac{1}{2}\sum_i \mu_i \sum_{j\sim i} \nabla^G_{ij} f \nabla^G_{ij} g$.

Developing $\mathcal{D}^G_\mu$ we get:

$$\begin{aligned}
\mathcal{D}^G_\mu(f,g) &= \frac{1}{2}\sum_i \mu_i \sum_{j\sim i} \nabla^G_{ij} f \nabla^G_{ij} g \\
&= \frac{1}{2}\sum_i \mu_i \sum_{j\sim i} (g(i) - g(j))(f(i) - f(j)) \\
&= \frac{1}{2}\sum_i \sum_{j\sim i} \frac{\mu_i S_{ij} + \mu_j S_{ji}}{2} \qquad \text{with } S_{ij} = \big(f(i) - f(j)\big)\big(g(i) - g(j)\big) \\
&= \frac{1}{2}\sum_i \sum_{j\sim i} \frac{S_{ij}(\mu_i + \mu_j)}{2} \qquad \text{using the symmetry properties of } S \\
&= \sum_{i\sim j} \frac{\mu_i + \mu_j}{2}\big(f(i) - f(j)\big)\big(g(i) - g(j)\big) \\
&= \frac{1}{2}\sum_i \sum_j A_{\mu,ij}\big(f(i) - f(j)\big)\big(g(i) - g(j)\big) \\
&= \frac{1}{2}\sum_i \sum_j g(i)A_{\mu,ij}\big(f(i) - f(j)\big) - g(j)A_{\mu,ij}\big(f(i) - f(j)\big) \\
&= \frac{1}{2}\sum_i \sum_j g(i)A_{\mu,ij}\big(f(i) - f(j)\big) - g(i)A_{\mu,ji}\big(f(j) - f(i)\big) \\
&= \sum_i \sum_j g(i)\, A_{\mu,ij}\, (f(i) - f(j)) \qquad \text{using the symmetry properties of } A_{\mu,ij} \\
&= \sum_i g(i)\left(D_{\mu,ii} f(i) - \sum_{j\sim i} A_{\mu,ij} f(j)\right)
\end{aligned}$$

where we have defined $A_\mu = M_\mu \odot A$, with $\odot$ being the Hadamard product and $M_{\mu,ij} = \frac{1}{2}(\mu_i + \mu_j)$. The weighted adjacency matrix A_μ preserves the original adjacency and the corresponding degree matrix is defined as usual $D_\mu = \mathrm{diag}\big(A_\mu \mathbf{1}\big)$. Finally, we immediately see that:

$$D^G_\mu(f,g) = (L_\mu f)^T g \Rightarrow L_\mu = D_\mu - A_\mu \tag{14}$$

Which allows us to just simply compute A_μ and D_μ.

□

C.4 Proof of Theorem 2

We derive the decomposition of L_μ:

$$\begin{aligned}(L_\mu f)_i &= \sum_{j\sim i} \frac{\mu_i + \mu_j}{2}(f(i) - f(j)) \\ &= \sum_{j\sim i} \left(\mu_i + \frac{\mu_j - \mu_i}{2}\right)(f(i) - f(j)) \\ &= \mu_i \sum_{j\sim i}(f(i) - f(j)) + \frac{1}{2}\sum_{j\sim i}(\mu_j - \mu_i)(f(i) - f(j)) \\ &= \mu_i (Lf)(i) - \frac{1}{2}\sum_{j\sim i} \nabla^G_{ij}\mu \, \nabla^G_{ij} f.\end{aligned}$$

□

C.5 Proof of Theorem 3

Since L_μ preserves the graph topology, its effect comes from changing the contribution of local differences $f(u) - f(v)$ (where u and v are two neighboring nodes) to the Dirichlet energy. This section makes this precise through Rayleigh quotients, derives bounds comparing the spectra of L and L_μ.

First, we compare L and L_μ through their Rayleigh quotients, $\mathcal{R}(f) = \frac{f^\top L f}{f^\top f}$, and $\mathcal{R}_\mu(f) = \frac{f^\top L_\mu f}{f^\top f}$. For $f \in \mathbb{R}^n$, define the normalized local variation profile $N(f) = \left(\frac{\|\nabla^G f(u)\|_2^2}{f^\top f}\right)_{u\in V}$, where $\|\nabla^G f(u)\|_2^2 = \sum_{v\sim u}\big(f(u) - f(v)\big)^2$, and the associated probability distribution $p_f = \frac{1}{\mathbf{1}^\top N(f)} N(f)$.

In order to complete the proof, let us first state a helpful lemma:

Lemma C.1 (Rayleigh quotient factorization)**.** *Let $f \in \mathbb{R}^n$. Then*

$$\mathcal{R}_\mu(f) = \|\mu\|_1 \, \mathbb{E}_\mu[p_f] \, \mathcal{R}(f),$$

where $\mathbb{E}_\mu[p_f]$ is the expectation of p_f under μ normalized to be a probability distribution.

Lemma C.1 shows that μ controls the spectrum by changing the cost of local variations: if μ is large where f varies strongly, then $\mathcal{R}_\mu(f)$ increases; if μ is small in those regions, it decreases. The spectrum of L_μ is therefore controlled by the global scale $\|\mu\|_1$, the spectrum of the underlying graph through $\mathcal{R}(f)$, and the alignment between μ and the local variation profile p_f.

We can now complete the proof of Theorem 3:

The Courant-Fischer min-max theorem gives the following characterization of the eigenvalues of L_μ: $\lambda_k^\mu = \min_{\substack{F\subset\mathbb{R}^n \\ \dim(F)=k+1}} \max_{\substack{f\in F \\ \|f\|_2=1}} R_\mu(f)$. Substituting the factorization of $R_\mu(f)$ from Lemma C.1 gives $\lambda_k^\mu = \min_{\dim(F)=k+1} \max_{\|f\|=1,\, f\in F} \Big(\|\mu\|_1 \, E_\mu[p_f] \, R(f)\Big)$, and since $\|\mu\|_1$ is independent of f, we obtain: $\lambda_k^\mu = \|\mu\|_1 \, \min_{\dim(F)=k+1} \max_{\|f\|=1,\, f\in F} \Big(E_\mu[p_f] \, R(f)\Big)$. We then have the following bound for any fixed subspace F of dimension $k+1$: $\lambda_k^\mu \le \|\mu\|_1 \max_{\|f\|=1,\, f\in F} \Big(E_\mu[p_f] \, R(f)\Big)$. Let us define the sets of subspaces: $\mathcal{F}_k = \big\{F : \dim(F) = k+1,\, \|f\| = 1,\, \max_{f\in F} R(f) = \lambda_k\big\}$.

Now choosing $F \in \mathcal{F}_k$ we obtain: $\lambda_k^\mu \le \lambda_k \|\mu\|_1 \max_{\|f\|=1,\, f\in F} \Big(E_\mu[p_f]\Big)$, and minimizing over all such choices yields: $\lambda_k^\mu \le \lambda_k \|\mu\|_1 \min_{F\in\mathcal{F}_k} \max_{\|f\|=1,\, f\in F} \Big(E_\mu[p_f]\Big)$.

Defining $C_k^\mu = \min_{F\in\mathcal{F}_k} \max_{|f|=1, f\in F} E_\mu[p_f]$, we finally obtain the first claimed upper bound: $\lambda_k^\mu \le \lambda_k |\mu|_1 C_k^\mu$. For the lower bound we first recall the dual Courant-Fischer max-min characterization $\lambda_k^\mu = \max_{\dim(S)=n-k} \min_{f\in S\ |f|=1} R_\mu(f)$. Then one can select the canonical subspace of eigenvectors

of L, $S_k = \mathrm{span}(u_k, \ldots, u_{n-1})$, and obtain $\lambda_k^\mu \geq \lambda_k |\mu|_1 \min_{f \in S_k \ |f|=1} E_\mu[p_f]$. Defining $c_k^\mu = \min_{\substack{f \in S_k \\ |f|=1}} E_\mu[p_f]$, gives the claimed lower bound: $\lambda_k^\mu \geq \lambda_k |\mu|_1 c_k^\mu$.

Theorem 3 shows that shaping μ gives targeted control over the spectrum, beyond a global rescaling by $\|\mu\|_1$. In particular, eigenvalues can be increased or decreased depending on how the potential overlaps with the local variation profiles of the corresponding eigenvectors.

□

C.6 Proof of Lemma C.1

We simply write:

$$\mathcal{R}_\mu(f) = \frac{1}{2}\,\mu^\top \cdot N(f) = \|\mu\|_1 \left(\frac{1}{\|\mu\|_1}\mu^\top \cdot p_f\right) \cdot \frac{1}{2}\left(\mathbf{1}^\top \cdot N(f)\right) = \|\mu\|_1\, \mathbb{E}_\mu[p_f]\, \mathcal{R}(f).$$

□

C.7 Eigenspace stability

Proposition 3. *Let E_r and E_r^μ be eigenspaces corresponding to the first $(r+1)$ eigenvalues of L and L_μ respectively. The $\sin\Theta$ distance between these subspaces is given by:*

$$\|\sin\Theta(E_r^\mu, E_r)\|_F^2 = \sum_{i=0}^{r} \sin^2\theta_i \leq \left(\frac{C}{c} - 1\right) \frac{\sum_{i=0}^{r} \lambda_i}{\lambda_{r+1} - \lambda_r},$$

where $c \geq \inf_{k\geq 0} c_k^\mu$ and $C = \sup_{0\leq k\leq r} C_r^\mu$ and θ_i are the principal angles between E_r and E_r^μ.

Proof. Let E_r and E_r^μ be eigenspaces corresponding to the first $(r+1)$ eigenvalues of L and L_μ respectively. The $\sin\Theta$ distance between these subspaces is given by:

$$\|\sin\Theta(E_r^\mu, E_r)\|_F^2 = \sum_{i=0}^{r} \sin^2\theta_i \leq \left(\frac{C}{c} - 1\right) \frac{\sum_{i=0}^{r} \lambda_i}{\lambda_{r+1} - \lambda_r},$$

where $c \geq \inf_{k\geq 0} c_k^\mu$ and $C = \sup_{0\leq k\leq r} C_r^\mu$.

By Courant–Fischer on the eigen subspaces $E_i = \mathrm{span}(u_0, \ldots, u_i) \subset E_r$ of L:

$$\lambda_i^\mu = \min_{\dim V = i+1} \max_{f\in V, f\neq 0} \frac{\langle f, L_\mu f\rangle}{\|f\|^2} \leq \max_{f\in E_i, f\neq 0} \frac{\langle f, L_\mu f\rangle}{\|f\|^2} \leq C\|\mu\|_1 \lambda_i,$$

using Theorem 3 and $C = \sup_{0\leq k\leq r} C_r^\mu$. Summing over the first $r+1$ modes gives the partial trace upper bound:

$$\sum_{i=0}^{r} \lambda_i^\mu \leq C\|\mu\|_1 \sum_{i=0}^{r} \lambda_i.$$

Now let $\{u_0^\mu, \ldots, u_r^\mu\}$ be an orthonormal eigenbasis of E_r^μ for L_μ and P_r^μ the corresponding orthogonal projector. We can lower bound the partial trace using Theorem 3:

$$\sum_{i=0}^{r} \lambda_i^\mu = \mathrm{Tr}(P_r^\mu L_\mu P_r^\mu) = \sum_{i=0}^{r} \langle u_i^\mu, L_\mu u_i^\mu\rangle \geq c\|\mu\|_1 \sum_{i=0}^{r} \langle u_i^\mu, L u_i^\mu\rangle = c\|\mu\|_1 \mathrm{Tr}(P_r^\mu L P_r^\mu),$$

where $c \geq \inf_{k\geq 0} c_k^\mu$.

To link these bounds with the subspace distance between E_r and E_r^μ, we split $P_r^\mu L P_r^\mu$ into orthogonal components relative to E_r:

$$\mathrm{Tr}(P_r^\mu L P_r^\mu) = \mathrm{Tr}(P_r^\mu (I - P_r) L (I - P_r) P_r^\mu) + \mathrm{Tr}(P_r^\mu P_r L P_r P_r^\mu),$$

where P_r is the orthogonal projector on E_r. Let $L|_{E_r^\perp} = (I - P_r)L(I - P_r)$ be the restriction of L to $E_r^\perp$, we have $L|_{E_r^\perp} \geq \lambda_{r+1} I$, and therefore:

$$\mathrm{Tr}(P_r^\mu (I - P_r) L (I - P_r) P_r^\mu) \geq \lambda_{r+1} \|(I - P_r) P_r^\mu\|_F^2 = \lambda_{r+1} \sum_{i=0}^{r} \sin^2\theta_i.$$

Likewise, we now deal with the complementary projector $P_r^\mu P_r L P_r P_r^\mu$. Let $Q = P_r P_r^\mu P_r$. It is self-adjoint on E_r with $0 \le Q \le I$ and $\mathrm{Tr}_{E_r}(I - Q) = \sum_{i=0}^{r} \sin^2\theta_i$. Rewriting the corresponding partial trace:

$$\mathrm{Tr}(P_r^\mu P_r L P_r P_r^\mu) = \mathrm{Tr}_{E_r}(QL|_{E_r}) = \mathrm{Tr}_{E_r}(L|_{E_r}) - \mathrm{Tr}_{E_r}((I-Q)L|_{E_r}).$$

Since $(I - Q) \ge 0$ and $L|_{E_r} \le \lambda_r I$ on E_r:

$$\mathrm{Tr}_{E_r}((I-Q)L|_{E_r}) \le \lambda_r \mathrm{Tr}_{E_r}(I-Q) = \lambda_r \sum_{i=0}^{r} \sin^2\theta_i,$$

and therefore:

$$\mathrm{Tr}(P_r^\mu P_r L P_r P_r^\mu) \ge \sum_{i=0}^{r} \lambda_i - \lambda_r \sum_{i=0}^{r} \sin^2\theta_i.$$

Combining the two terms yields:

$$\mathrm{Tr}(P_r^\mu L P_r^\mu) \ge \sum_{i=0}^{r} \lambda_i + (\lambda_{r+1} - \lambda_r) \sum_{i=0}^{r} \sin^2\theta_i.$$

Finally, combining the upper and lower bounds gives us:

$$c\|\mu\|_1 \left(\sum_{i=0}^{r} \lambda_i + (\lambda_{r+1} - \lambda_r) \sum_{i=0}^{r} \sin^2\theta_i \right) \le \sum_{i=0}^{r} \lambda_i^\mu \le C\|\mu\|_1 \sum_{i=0}^{r} \lambda_i.$$

Dividing by $c\|\mu\|_1$ and rearranging term provides the announced substance distance formula:

$$\|\sin\Theta(E_r^\mu, E_r)\|_F^2 = \sum_{i=0}^{r} \sin^2\theta_i \le \left(\frac{C}{c} - 1\right) \frac{\sum_{i=0}^{r} \lambda_i}{\lambda_{r+1} - \lambda_r}.$$

□

C.8 Proof of the Nonlinear Analysis 4.2

To rigorously bound the dynamics of the linearized output, we must bridge the unweighted edge-wise action of the point-wise non-linearity with the μ-weighted spectral geometry. Because the spatially varying Jacobian $D = \mathrm{diag}(\sigma'(\mathbf{z}))$ does not commute with the unweighted graph Laplacian L, an input perturbation $\mathbf{h}$ band-limited to a spectral subspace F_k of L will get scattered. The output $\mathbf{y} = D\mathbf{h}$ will therefore contain both an in-band component $\mathbf{y}_k \in F_k$ and an out-of-band leakage component $\mathbf{y}_\perp \in F_k^\perp$.

Theorem 5 (Complete Spectral Energy Bound for Linearized Dynamics)**.** *Let a spectral μ-GNN layer evolve under a point-wise non-linear activation σ satisfying the following conditions:*

1. *1-Lipschitz Continuity:* $0 \le \sigma'(x) \le 1$.
2. *Strict Non-Saturation:* $\inf \sigma'(x) = c > 0$.
3. *Bounded Curvature: σ' is β-Lipschitz.*

Let $\mathbf{z}$ be the pre-activation state, and let $\mathbf{h} \in F_k$ be an infinitesimal tangent perturbation residing strictly in the k-th eigenspace of the unweighted Laplacian L. Let $\mathbf{y} = D\mathbf{h}$ be the linearized output, where $D = diag(\sigma'(\mathbf{z}))$.

Assume Theorem 3 holds, providing local equivalence constants $0 < c_k^\mu \le C_k^\mu < \infty$ for vectors in F_k, and global bounds $0 < c_{\min}^\mu \le C_{\max}^\mu$ across the entire spectrum.

Then, for any arbitrary parameters $\gamma, \eta > 0$, the μ-weighted Dirichlet energy of the output, $\mathcal{E}_\mu(\mathbf{y})$, satisfies the following bounds:

Upper Bound:

$$\mathcal{E}_\mu(\mathbf{y}) \le (1+\eta)(1+\gamma) \frac{C_k^\mu}{c_k^\mu} \mathcal{E}_\mu(\mathbf{h}) + R^\mu(\mathbf{h}, \mathbf{z}, \gamma, \eta)$$

Lower Bound:

$$\mathcal{E}_\mu(\mathbf{y}) \geq \frac{(1-\eta)c^2 c_k^\mu}{(1+\gamma)C_k^\mu}\mathcal{E}_\mu(\mathbf{h}) - r^\mu(\mathbf{h},\mathbf{z},\gamma,\eta)$$

Proof. We first decompose the linearized output $\mathbf{y}$ orthogonally with respect to the standard unweighted inner product. Let P_k be the orthogonal projection operator onto F_k:

$$\mathbf{y} = P_k\mathbf{y} + (I - P_k)\mathbf{y} = \mathbf{y}_k + \mathbf{y}_\perp.$$

Because P_k commutes with the unweighted Laplacian L, the unweighted Dirichlet energy perfectly decouples:

$$\mathcal{E}(\mathbf{y}) = \mathbf{y}^T L\mathbf{y} = \mathcal{E}(\mathbf{y}_k) + \mathcal{E}(\mathbf{y}_\perp).$$

We analyze the total unweighted energy $\mathcal{E}(\mathbf{y}) = \mathcal{E}(D\mathbf{h})$ by expanding the edge-wise differences: $(y_i - y_j) = d_i(h_i - h_j) + h_j(d_i - d_j)$. Applying Young's inequality $(a+b)^2 \leq (1+\gamma)a^2 + (1+1/\gamma)b^2$ and summing over all edges yields the upper bound on the total unweighted energy:

$$\mathcal{E}(\mathbf{y}) \leq (1+\gamma)\mathcal{E}(\mathbf{h}) + \underbrace{\left(1+\frac{1}{\gamma}\right)\beta^2\|\mathbf{h}\|_\infty^2\mathcal{E}(\mathbf{z})}_{\mathcal{S}(\mathbf{h},\mathbf{z})},$$

where $\mathcal{S}$ is the purely spatial scattering term. Symmetrically, applying $(a+b)^2 \geq \frac{1}{1+\gamma}a^2 - \frac{1}{\gamma}b^2$ alongside the non-saturating condition $d_i \geq c$ yields the unweighted lower bound:

$$\mathcal{E}(\mathbf{y}) \geq \frac{c^2}{1+\gamma}\mathcal{E}(\mathbf{h}) - \frac{1}{\gamma}\beta^2\|\mathbf{h}\|_\infty^2\mathcal{E}(\mathbf{z}). \tag{15}$$

Since L_μ does not necessarily commute with L, the μ-weighted energy contains a cross-term:

$$\mathcal{E}_\mu(\mathbf{y}) = \mathcal{E}_\mu(\mathbf{y}_k) + 2\mathbf{y}_k^T L_\mu \mathbf{y}_\perp + \mathcal{E}_\mu(\mathbf{y}_\perp).$$

By applying Young's inequality to the positive semi-definite bilinear form induced by L_μ, we bound the cross-terms for any $\eta > 0$:

$$\mathcal{E}_\mu(\mathbf{y}) \leq (1+\eta)\mathcal{E}_\mu(\mathbf{y}_k) + \left(1+\frac{1}{\eta}\right)\mathcal{E}_\mu(\mathbf{y}_\perp).$$

Because $\mathbf{y}_k \in F_k$ and $\mathbf{y}_\perp \in F_k^\perp$, we can now apply the bounds from Theorem 3:

$$\mathcal{E}_\mu(\mathbf{y}_k) \leq C_k^\mu\mathcal{E}(\mathbf{y}_k) \quad \text{and} \quad \mathcal{E}_\mu(\mathbf{y}_\perp) \leq C_{\max}^\mu\mathcal{E}(\mathbf{y}_\perp),$$

where $C_{\max}^\mu = \sup_k C_k^\mu$. Substituting these into the μ-energy bound, we get:

$$\mathcal{E}_\mu(\mathbf{y}) \leq (1+\eta)C_k^\mu\mathcal{E}(\mathbf{y}_k) + \left(1+\frac{1}{\eta}\right)C_{\max}^\mu\mathcal{E}(\mathbf{y}_\perp).$$

Since $\mathcal{E}(\mathbf{y}_k) \leq \mathcal{E}(\mathbf{y})$, we can replace the in-band energy with the total unweighted energy bound:

$$\mathcal{E}_\mu(\mathbf{y}) \leq (1+\eta)C_k^\mu\big[(1+\gamma)\mathcal{E}(\mathbf{h}) + \mathcal{S}(\mathbf{h},\mathbf{z})\big] + \left(1+\frac{1}{\eta}\right)C_{\max}^\mu\mathcal{E}(\mathbf{y}_\perp).$$

Finally, assuming that the Davis–Kahan conditions are satisfied so that the corresponding eigenspaces of L and L_μ can be identified up to a small perturbation, we apply Theorem 3 to the input perturbation $\mathbf{h} \in F_k$. Assuming for simplicity that $|\mu|_1 = 1$, this yields $\mathcal{E}(\mathbf{h}) \leq \frac{1}{c_k^\mu}\mathcal{E}_\mu(\mathbf{h})$ and the leading mode-preserving term becomes exactly:

$$(1+\eta)(1+\gamma)\frac{C_k^\mu}{c_k^\mu}\mathcal{E}_\mu(\mathbf{h}).$$

Let us now bound the second term $\mathcal{E}(\mathbf{y}_\perp)$: Let $d_i = \sigma'(z_i)$. Because the input perturbation $\mathbf{h} \in F_k$, scaling it by any global scalar constant $\bar{d} \in \mathbb{R}$ ensures the resulting vector remains strictly inside the eigenspace. Therefore, $(I - P_k)(\bar{d}\mathbf{h}) = \mathbf{0}$. We can freely subtract this shifted in-band vector before projecting: $\mathbf{y}_\perp = (I - P_k)(D\mathbf{h}) = (I - P_k)\big((D - \bar{d}I)\mathbf{h}\big)$. Since P_k is an orthogonal

projector with respect to the unweighted Dirichlet inner product, the residual operator $(I - P_k)$ is non-expansive. Thus, the energy of the projection is upper-bounded by the energy of the shifted vector: $\mathcal{E}(\mathbf{y}_\perp) \le \mathcal{E}\big((D - \bar{d}I)\mathbf{h}\big)$. The unweighted Dirichlet energy of the shifted vector is written explicitly as: $\mathcal{E}\big((D - \bar{d}I)\mathbf{h}\big) = \frac{1}{2}\sum_{i,j} A_{ij}\big((d_i - \bar{d})h_i - (d_j - \bar{d})h_j\big)^2$.We decompose the edge-wise difference into $(d_i - \bar{d})h_i - (d_j - \bar{d})h_j = (d_i - \bar{d})(h_i - h_j) + h_j(d_i - d_j)$ and applying Young's inequality $(a+b)^2 \le (1+\alpha)a^2 + \big(1 + \frac{1}{\alpha}\big)b^2$ for any parameter $\alpha > 0$, we bound the bound the Dirichlet energy as follows: $\mathcal{E}\big((D - \bar{d}I)\mathbf{h}\big) \le (1+\alpha)\underbrace{\frac{1}{2}\sum_{i,j} A_{ij}(d_i - \bar{d})^2(h_i - h_j)^2}_{I_1} + \big(1 + \frac{1}{\alpha}\big)\underbrace{\frac{1}{2}\sum_{i,j} A_{ij}h_j^2(d_i - d_j)^2}_{I_2}$

We can bound I_2 using the supremum norm of the perturbation ($\|\mathbf{h}\|_\infty$) and the β-Lipschitz property of the activation derivative $(d_i - d_j)^2 \le \beta^2(z_i - z_j)^2$: $I_2 \le \beta^2\|\mathbf{h}\|_\infty^2\big(\frac{1}{2}\sum_{i,j} A_{ij}(z_i - z_j)^2\big) = \beta^2\|\mathbf{h}\|_\infty^2\mathcal{E}(\mathbf{z})$. To bound I_1, we choose the global constant $\bar{d}$ as the spatial mean of the activation gate across the graph: $\bar{d} = \frac{1}{N}\sum_{v=1}^{N} d_v$. The maximum deviation from this mean is bounded by the maximum pairwise difference on the graph: $(d_i - \bar{d})^2 \le \max_{u,v}(d_u - d_v)^2 \le \beta^2 \max_{u,v}(z_u - z_v)^2$. Let $\mathcal{P}_{uv}$ be the shortest path between nodes u and v, with path length bounded by the graph diameter, $L(\mathcal{P}_{uv}) \le \mathrm{diam}(G)$. Expressing the difference $z_u - z_v$ as a telescoping sum along the edges $e \in \mathcal{P}_{uv}$ and applying the Cauchy-Schwarz inequality yields: $(z_u - z_v)^2 = \big(\sum_{e\in\mathcal{P}_{uv}}(z_{e^+} - z_{e^-})\big)^2 \le L(\mathcal{P}_{uv})\sum_{e\in\mathcal{P}_{uv}}(z_{e^+} - z_{e^-})^2 \le \mathrm{diam}(G)\sum_{(i,j)\in E}(z_i - z_j)^2$, where the sum over all edges is exactly $2\mathcal{E}(\mathbf{z})$. We therefore have: $(d_i - \bar{d})^2 \le 2\beta^2\mathrm{diam}(G)\mathcal{E}(\mathbf{z})$. Substituting this uniform upper bound back into I_1 gives: $I_1 \le 2\beta^2\mathrm{diam}(G)\mathcal{E}(\mathbf{z})\big(\frac{1}{2}\sum_{i,j} A_{ij}(h_i - h_j)^2\big) = 2\beta^2\mathrm{diam}(G)\mathcal{E}(\mathbf{h})\mathcal{E}(\mathbf{z})$. Combining the bounds for I_1 and I_2 yields the strict upper bound on the unweighted energy of the out-of-band spectral leakage: $\mathcal{E}(\mathbf{y}_\perp) \le \big[(1+\alpha)\cdot 2\mathrm{diam}(G)\mathcal{E}(\mathbf{h}) + \big(1 + \frac{1}{\alpha}\big)\|\mathbf{h}\|_\infty^2\big]\beta^2\mathcal{E}(\mathbf{z})$.

Finally, we consolidate this term and the previous scattering $\mathcal{S}(\mathbf{h}, \mathbf{z})$ into the upper roughness penalty $R^\mu(\mathbf{h}, \mathbf{z}, \gamma, \eta)$.

The lower bound follows an identical algebraic structure. Bounding the L_μ cross-terms from below gives:

$$\mathcal{E}_\mu(\mathbf{y}) \ge (1 - \eta)\mathcal{E}_\mu(\mathbf{y}_k) - \left(\frac{1}{\eta} - 1\right)\mathcal{E}_\mu(\mathbf{y}_\perp).$$

Applying Theorem 3 and utilizing $\mathcal{E}(\mathbf{y}_k) = \mathcal{E}(\mathbf{y}) - \mathcal{E}(\mathbf{y}_\perp)$ allows us to lower-bound the leading term strictly by the unweighted total energy $\mathcal{E}(\mathbf{y})$, to which we apply the lower quadratic bound (15). Mapping back via $\mathcal{E}(\mathbf{h}) \ge \frac{1}{C_k^\mu}\mathcal{E}_\mu(\mathbf{h})$ yields the required geometric condition number, while we consolidate the residual subtractions into r^μ.

□

Corollary 1 (Recovery of the Stylized Bound)**.** *Let the spatial regularity $\mathcal{E}(z)$ of the activation profile be bounded by $\varepsilon \ge 0$. By setting the arbitrary parameters γ and η according to ε, the rigorous bounds of Theorem 4 simplify to the stylized bounds.*

Proof. The upper bound in Theorem 4 relies on two free tuning parameters: $\gamma > 0$ (controlling edge-wise spatial scattering) and $\eta > 0$ (controlling cross-subspace spectral leakage). Using the stated hypothesis on spatial roughness gives:

$$\mathcal{E}_\mu(\mathbf{y}) \le (1+\eta)(1+\gamma)\frac{C_k^\mu}{c_k^\mu}\mathcal{E}_\mu(\mathbf{h}) + \left[(1+\eta)C_k^\mu\left(1 + \frac{1}{\gamma}\right) + \left(1 + \frac{1}{\eta}\right)C_{\max}^\mu\right]\mathcal{O}(\varepsilon).$$

We couple γ and η directly to this quantity by setting $\gamma = \sqrt{\varepsilon}$ and $\eta = \sqrt{\varepsilon}$ and observe:

$$(1+\sqrt{\varepsilon})C_k^\mu\left(1 + \frac{1}{\sqrt{\varepsilon}}\right)\mathcal{O}(\varepsilon) = (1+\sqrt{\varepsilon})C_k^\mu\big(\mathcal{O}(\varepsilon) + \mathcal{O}(\sqrt{\varepsilon})\big) = \mathcal{O}(\sqrt{\varepsilon}).$$

The out-of-band leakage yields:

$$\left(1 + \frac{1}{\sqrt{\varepsilon}}\right)C_{\max}^\mu\mathcal{O}(\varepsilon) = C_{\max}^\mu\big(\mathcal{O}(\varepsilon) + \mathcal{O}(\sqrt{\varepsilon})\big) = \mathcal{O}(\sqrt{\varepsilon}).$$

Absorbing the constants $C_k^\mu, C_{\max}^\mu$, the sum of both error terms remains exactly $\mathcal{O}(\sqrt{\varepsilon})$. Substituting these into the bounds from Theorem 4 yields the asymptotic form:

$$\left(1-\mathcal{O}(\sqrt{\varepsilon})\right) m^2 \frac{c_k^\mu}{C_k^\mu}\mathcal{E}_\mu(\mathbf{h}) - \mathcal{O}(\sqrt{\varepsilon}) \le \mathcal{E}_\mu(\mathbf{y}) \le \left(1+\mathcal{O}(\sqrt{\varepsilon})\right) \frac{C_k^\mu}{c_k^\mu}\mathcal{E}_\mu(\mathbf{h}) + \mathcal{O}(\sqrt{\varepsilon}).$$

In the limit as the activation gate becomes perfectly smooth ($\varepsilon \to 0$), the tuning parameters $\gamma, \eta \to 0$ simultaneously, and we recover the idealized spectral corridor presented in the stylized version of Theorem 4:

$$m^2 \frac{c_k^\mu}{C_k^\mu} \;\le\; \frac{\mathcal{E}_\mu(\mathbf{y})}{\mathcal{E}_\mu(\mathbf{h})} \;\le\; M^2 \frac{C_k^\mu}{c_k^\mu}.$$

□

D Details on Chebnet Architecture 3.3

ChebNet is particularly well suited to this perspective. It defines graph convolutions through spectral filters of the Laplacian, while avoiding the cost of an explicit eigendecomposition through a polynomial approximation. ChebNet approximates this filter using Chebyshev polynomials, yielding

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K} \Theta_k T_k(\widetilde{\Lambda}), \qquad \widetilde{\Lambda} = \frac{2\Lambda}{\lambda_{\max}} - I, \tag{16}$$

where T_k is the k-th Chebyshev polynomial and $\widetilde{\Lambda}$ rescales the spectrum to the interval $[-1, 1]$. Since polynomials commute with the eigendecomposition, the resulting layer can be written directly in terms of the Laplacian:

$$y = \sum_{k=0}^{K} T_k(\widetilde{L}) x \Theta_k, \qquad \widetilde{L} = \frac{2L}{\lambda_{\max}} - I. \tag{17}$$

This form is crucial: the filter depends only on repeated applications of the Laplacian, and can therefore be evaluated in $O(Km)$ time without computing the eigenvectors explicitly Defferrard et al. (2016).

E Details on Section 4.1

We now introduce the geometric construction underlying our approach. Rather than adding an explicit advection field to a diffusion equation, we modify the diffusion geometry itself through a positive weight μ. This distinction is important on graphs. In the continuous setting, a term such as $v \cdot \nabla f$ for a scalar function f and v a vector field, is well defined because both $v(x)$ and $\nabla f(x)$ live in the same tangent space. On a graph, however, node signals live on vertices, whereas discrete gradients naturally live on edges. Defining a vector field therefore requires additional non-canonical choices, such as an orientation. Dirichlet forms and Laplacians, in contrast, admit direct graph analogues. This motivates inducing a transport bias through a weighted diffusion geometry rather than by prescribing an explicit advective field.

We first recall the unweighted continuous case. Let $f, g : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ be smooth functions vanishing on $\partial\Omega$. The standard Dirichlet form is $\mathcal{D}(f,g) = \frac{1}{2}\int_\Omega \nabla f(x) \cdot \nabla g(x) dx$. By integration by parts, using the boundary condition, one obtains $\mathcal{D}(f,g) = -\frac{1}{2}\int_\Omega g(x)\Delta f(x)\, dx$. Hence, with respect to the standard $L^2(dx)$ inner product, the associated positive operator is $Lf = -\frac{1}{2}\Delta f$. The corresponding gradient flow is the usual heat equation $\partial_t f = -Lf = \frac{1}{2}\Delta f$. Thus, the standard Dirichlet form recovers the usual Laplacian and ordinary diffusion. We now introduce a positive weight $\mu : \Omega \to \mathbb{R}_{>0}$ and define the weighted Dirichlet form $\mathcal{D}_\mu(f,g) = \frac{1}{2}\int_\Omega \mu(x)\nabla f(x) \cdot \nabla g(x) dx$. This form penalizes local variations differently depending on the position. Equivalently, μ changes the geometry in which diffusion takes place.

There are two useful ways of representing the operator associated with the same Dirichlet form. First, using the weighted inner product $\langle f, g\rangle_\mu = \int_\Omega f(x)g(x)\mu(x)\, dx$, integration by parts gives $\mathcal{D}_\mu(f,g) = \langle g, L_\mu f\rangle_\mu$, where

$$L_\mu f = -\frac{1}{2\mu}\nabla \cdot (\mu \nabla f) = -\frac{1}{2}\Delta f - \frac{1}{2}\nabla \log \mu \cdot \nabla f. \tag{18}$$

The second term in (18) is first order and therefore has the form of a drift or advective contribution. However, it is not introduced through an independently prescribed velocity field. It is induced by the spatial variation of the weight μ.

Equivalently, one may represent the same Dirichlet form using the standard $L^2(dx)$ inner product. This gives

$$\widetilde{L}_\mu f = -\frac{1}{2}\nabla\cdot(\mu\nabla f) = -\frac{1}{2}\mu\Delta f - \frac{1}{2}\nabla\mu\cdot\nabla f. \tag{19}$$

The associated diffusion equation is $\partial_t f = -\widetilde{L}_\mu f$. This second representation gives the intuition of a diffusion with spatially varying diffusivity μ. Expanding the weighted diffusion operator in (19) produces the first-order term $\nabla\mu\cdot\nabla f$, which acts as a drift-like bias. It appears because the diffusion weight itself varies in space. This is the continuous mechanism that we now transfer to graphs.

In the unweighted graph case, the Dirichlet form is $\mathcal{D}^G(f,g) = \frac{1}{2}\sum_{i,j} A_{ij}(f_i - f_j)(g_i - g_j)$. Using the symmetry of A, this rewrites as $\mathcal{D}^G(f,g) = \sum_i g_i \sum_j A_{ij}(f_i - f_j) = g^\top(D - A)f$. Thus, as in the continuous case, the operator represented by the unweighted graph Dirichlet form is the combinatorial Laplacian $L = D - A$.

F Link with spectral bounding constants

Now, consider Theorem 3. It shows that

$$\lambda_k\|\mu\|_1 c_k^\mu \le \lambda_k^\mu \le \lambda_k\|\mu\|_1 C_k^\mu,$$

and therefore the gap is controlled by the ratios of the spectral constants

$$\frac{c_k^\mu}{C_1^\mu}\rho_k \le \rho_k^\mu \le \frac{C_k^\mu}{c_1^\mu}\rho_k,$$

which also shows that these constants cannot be universal precisely because they dictate how the potential is allowed to control this gap. The constants c_k^μ, C_k^μ are not nuisance constants; they are the mechanism by which μ selectively changes modal decay.

Further, the definition of these constants illuminates what the potential μ can actually do. We show $R_\mu(f) = |\mu|_1\mathbb{E}_\mu[p_f]R(f)$, where p_f is the normalized local-variation profile of f (a probability distribution over vertices, putting mass on where f varies) and $\mathbb{E}_\mu[p_f] = \frac{1}{|\mu|_1}\sum_i \mu_i p_f(i)$ is the overlap between μ and that profile. The constants in Theorem 3 are thus the extremal values of this overlap term as f ranges over (a neighborhood of) the k-th eigenspace:

$$c_k^\mu = \min_{f\in F_k,\ |f|=1} \mathbb{E}_\mu[p_f], \qquad C_k^\mu = \max_{f\in F_k,\ |f|=1} \mathbb{E}_\mu[p_f].$$

So c_k^μ, C_k^μ encode how μ overlaps with the local variation profile of the k-th mode. Now consider what this means for a SBM (or topological bottleneck) example:

- The Fiedler mode u_1 has its variation concentrated on the bridge edge (that's what makes it the community-separating mode). If μ is small on/around the bottleneck, p_{u_1} puts most of its mass exactly where μ is small, so $\mathbb{E}_\mu[p_{u_1}]$ is small — both c_1^μ and C_1^μ shrink, and λ_1^μ is suppressed relative to λ_1. This leads to a "community-separating mode survives longer" effect.
- Bulk/high-frequency modes vary within each community, i.e. where μ is large. Their p_f overlaps heavily with the high-μ region, so $\mathbb{E}_\mu[p_f]$ is large, c_k^μ, C_k^μ are large, and those eigenvalues are inflated — they decay faster under L_μ than under L.

G Experimental setup

G.1 Hyper-parameters

We report in Tables 2 and 3 the hyper-parameters used in Section 6.

Table 2: Hyper-parameters on `ogbn-proteins`

Hyper-parameter	**Values**
Hidden dimension d	55
Polynomial order K	5
Number of layers	1-15
Activation function	tanh
Learning rate	0.001
Optimizer	Adam
Number of parts	40 train set, 5 test set (for Pytorch Geometric RandomNodeLoader)
Post-layer block	BatchNorm + Dropout (p = 0.2 first layer, p = 0.1 thereafter; omitted before decoder)

Table 3: Hyper-parameter on Oversquashing on Barbell.

Hyper-parameter	**Values**
Hidden dimension d	128
Polynomial order K	7
Num of layers	1-20
Batch size	64
Learning rate	0.001
Optimizer	Adam
Activation function	Relu
Post-layer block	BatchNorm + Dropout (p = 0.2 first layer, p = 0.1 thereafter; omitted before decoder)
Scheduler	ReduceLROnPlateau (Torch learning rate scheduler), factor 0.5, patience 10

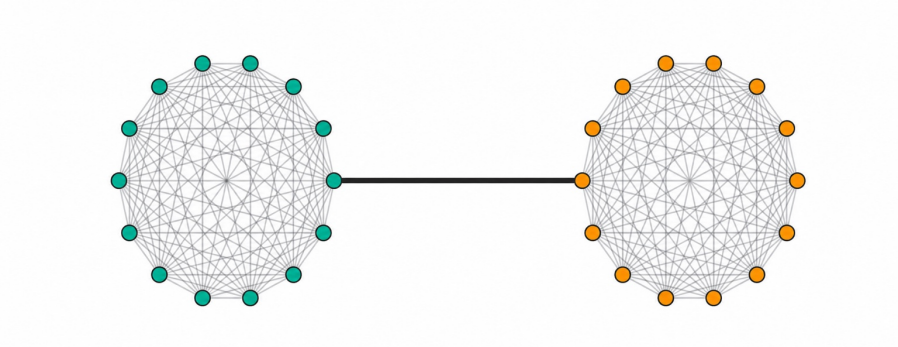


Figure 10: Illustration of a Barbell graph composed of two densely connected node clusters

G.2 Barbell Task description for Resistance Distance (Experiment in Figure 3)

The Barbell dataset is designed to test whether a GNN can transmit information across a severe graph bottleneck. Each graph consists of two densely connected complete subgraphs, called the "bells," joined by a narrow path or "bridge" as shown in Figure 10. Nodes inside each bell are highly connected, while the bridge is the only route through which information can pass from one side of the graph to the other. The learning task is node-level regression. Only two designated nodes carry a signal, one node from clique 1 and one node from clique 2. They are picked once and fixed across all samples. Their target labels consist in the other clique node signal. Every other node's feature and target are pure noise, unrelated to anything. The task isolates whether mu emphasizes the path between two designated far-apart nodes, helping message propagation.

G.3 Barbell Task Description for Evaluation (Experiment in Figure 4)

The learning task is node-level regression: each node must predict the average input feature of the nodes in the opposite bell. This makes the dataset a direct probe of long-range communication,

Table 4: Barbell dataset configurations used in the experiments.

Parameter	**Depth experiment Figure 4**	**Figure 3**
Number of nodes	100 (50 per clique)	40
Training samples	200	200
Validation samples	43	43
Test samples	43	43
Cross-clique mean	$\pm\sqrt{3 \cdot 50}$	–
Standard deviation	$\sqrt{50}$	–
Label value difference	–	4.0
Feature noise standard deviation	–	0.5
Label noise standard deviation	–	0.03

because successful prediction requires information to cross the bridge rather than remain trapped within one dense cluster. Performance is measured using mean squared error (MSE), where high errors indicate either over-squashing, when information from the opposite bell fails to pass through the bridge, or over-smoothing, when node representations collapse and lose local distinctions.