Long-range Propagation in Graph Neural Networks Benefits from Learnable Geometry and Unitary Operators

Pierre-Gabriel Berlureau[1,2]

[1]Department of Computer Science, École Normale Supérieure, PSL University
[2]Signal Processing Laboratory LTS2, EPFL

July 2025

Abstract

Graph Neural Networks (GNNs) are deep learning models designed to process graph-structured data by propagating information across nodes. Despite their success, they are known to exhibit oversmoothing and oversquashing, which can cause representational collapse in deep networks and reduce sensitivity to information from distant nodes. In this work, conducted under the supervision of Pierre Vandergheynst, we analyze these limitations within a broad class of Message-Passing Neural Networks (MPNNs), showing in a simplified version of the models that both phenomena arise from common spectral and dynamical mechanisms. To address them, we propose two architectures inspired by quantum dynamics and biased diffusion, along with theoretical analyses and empirical evaluations demonstrating how the proposed architectures mitigate these shortcomings.

1 Introduction

Traditional deep learning methods are tailored to Euclidean data structures, such as feature vectors or images. However, many real-world systems like social networks or molecular structures are inherently relational and better modeled as graphs, where nodes represent entities and edges represent interactions. Effectively analyzing such systems requires learning from both node features and the underlying topology. This motivates the development of algorithms that operate directly on graphs, integrating structural and attribute information.

Graph Neural Networks (GNNs) (Bruna et al., 2014; Defferrard et al., 2017; Kipf and Welling, 2017; Gilmer et al., 2017) have become a core tool for learning on graph-structured data. Modern architectures, including GAT (Veličković et al., 2018), GraphSAGE (Hamilton et al., 2017), and GIN (Xu et al., 2019), build on the message-passing paradigm to aggregate information from local neighborhoods. However, as their depth increases, GNNs face fundamental limitations such as *vanishing gradients* (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997; Pascanu et al., 2013), *oversmoothing* (NT and Maehara, 2019; Cai and Wang, 2020; Rusch et al., 2023), and *oversquashing* (Alon and Yahav, 2021; Topping et al., 2021; Giovanni et al., 2023), hindering their ability to capture interactions between very distant nodes.

The first part of this work, conducted during the initial three months of my internship under the sole supervision of Pierre Vandergheynst, builds on recent advances (Arroyo et al., 2025; Arnaiz-Rodriguez and Errica, 2025; Park et al., 2025) to show that oversmoothing and oversquashing stem from the same underlying spectral and dynamical phenomena. We characterize these effects within a broad class of Message-Passing Neural Networks (MPNNs) and introduce two architectures based on quantum dynamics and biased diffusion to mitigate them. The second part, in collaboration with Victor Kawasaki-Borruat and still under the supervision of Pierre Vandergheynst, improves upon these models by proposing a ChebNet-inspired architecture (Defferrard et al., 2017) using Chebyshev polynomials to reduce computational complexity.

2 Setup and notations

Consider an undirected graph $G = (V, E)$ with $n = |V|$ nodes, $m = |E|$ edges, an adjacency matrix $A \in \mathbb{R}^{n \times n}$ and a diagonal degree matrix $D \in \mathbb{R}^{n \times n}$ such that $D_{u,u}$ is the degree of node u (denoted by d_u). Each node $u \in V$ has an associated feature vector $X_u \in \mathbb{R}^d$, collected in the feature matrix $X \in \mathbb{R}^{n \times d}$. A Graph Neural Network (GNN) is a parameterized function f_θ mapping (G, X) to predictions $Y \in \mathbb{R}^{d'}$, trained by minimizing a loss $\mathcal{L}$ over labeled data.

Most modern GNNs follow the Message Passing Neural Network (MPNN) framework (Gilmer et al., 2017), where node representations are iteratively updated through K layers by aggregating information from neighbors. At layer k, the node embedding $X_u^{(k)}$ is computed as

$$X_u^{(k)} = \phi^{(k)}\left(X_u^{(k-1)}, \psi^{(k)}\left(\{X_v^{(k-1)} : v \in \mathcal{N}(u)\}\right)\right) \quad \text{and} \quad X_u^{(0)} = X_u,$$

where $\phi^{(k)}$ is a learnable update function and $\psi^{(k)}$ is a permutation-invariant aggregation (e.g., sum or mean) over the neighborhood $\mathcal{N}(u)$ of node u.

A widely used instance of MPNNs is the Graph Convolutional Network (GCN) (Kipf and Welling, 2017), which performs layer-wise updates as

$$X^{(k)} = \sigma\left(\hat{A} X^{(k-1)} W^{(k-1)}\right),$$

where $\hat{A} = (D + I)^{-1/2}(A + I)(D + I)^{-1/2}$ is the symmetrically normalized adjacency matrix with added self-loops, $W^{(k-1)}$ are learnable weights, and σ is a pointwise nonlinearity. Following standard notations, we denote $D + I$ as $\tilde{D}$ and $A + I$ as $\tilde{A}$.

Matrix Product Notation For a sequence of matrices $\{M^{(i)}\}$, we denote the ordered product from index i to j as

$$M_{i \to j} = M^{(i)} M^{(i+1)} \cdots M^{(j)} \quad \text{for } i < j,$$

and conversely,

$$M_{i \leftarrow j} = M^{(i)} M^{(i-1)} \cdots M^{(j)} \quad \text{for } i > j.$$

Unless otherwise specified, all graphs considered in this work are connected and undirected. Proofs of theoretical results are deferred to the Appendix.

3 Long-range dependencies

This section builds on the analysis of Arroyo et al. (2025). We nuance the conclusion of its Theorem 3.2 on the tendency of classical message passing GNNs to worsen the training stability compared to classic sequence models and propose a unified perspective of oversmoothing and oversquashing, identifying them as two manifestations of the same phenomena.

3.1 Vanishing Gradients

The vanishing gradient problem (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997; Pascanu et al., 2013), characterized by exponentially diminishing gradients during backpropagation, poses a significant challenge for training deep neural networks. In Graph Convolutional Networks (GCNs), this issue naturally arises from their recursive layer design, analogous to Recurrent Neural Networks (RNNs), where repeated graph propagation and linear transformations can lead to gradient decay and numerical instability. However, we show that GCNs do not inherently suffer worse gradient propagation than RNNs. To formalize this, we analyze a simplified linear GCN model without nonlinearities using the update rule:

$$X^{(k)} = \hat{A} X^{(k-1)} W^{(k)}$$

Let us consider the gradient of the loss function $\mathcal{L}$ with respect to parameters of an earlier layer $W^{(l)}$. This gradient appears in the update rule of standard gradient descent:

$$(W^{(l)})' = W^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W^{(l)}}$$

where η is the learning rate. Using the chain rule, we expand the gradient as:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \frac{\partial \mathcal{L}}{\partial X^{(L)}} \cdot J_{L \leftarrow l+1} \cdot \frac{\partial X^{(l)}}{\partial W^{(l)}}.$$

where $J^{(k)} = \frac{\partial X^{(k)}}{\partial X^{(k-1)}}$. By vectorizing matrices (i.e., stacking columns into a vector), we have:

$$J^{(k)} = \frac{\partial \operatorname{vec}(X^{(k)})}{\partial \operatorname{vec}(X^{(k-1)})} = (W^{(k)})^\top \otimes \hat{A},$$

where $\otimes$ denotes the Kronecker product. This follows from the matrix identity $\operatorname{vec}(ABC) = (C^\top \otimes A)\operatorname{vec}(B)$. Thus, the squared Jacobian norm at layer l is driven by:

$$\|J_{L \leftarrow l+1}\|_2^2 = \operatorname{Tr}((W_{L \leftarrow l+1})^\top W_{L \leftarrow l+1}) \cdot \operatorname{Tr}(\hat{A}^{2(L-l)})$$

The vanishing gradients phenomenon is characterized by an exponential decay of this norm. However, the following proposition frames the graph's contribution in the above quantity, nuancing the conclusion of Theorem 3.2 of Arroyo et al. (2025).

Proposition 3.1. *Let $\hat{A}$ be the symmetrically normalized adjacency matrix of a connected and unidrected graph with added self-loops:*

$$\frac{n}{d_{\max}+1} \leq \operatorname{Tr}(\hat{A}^k) \leq \frac{n}{d_{\min}+1} \quad \forall k \in 1, 2, \dots$$

Moreover, this framing is tight in the case of d-regular graphs.

As this framing is uniform in k, it shows that the graph operator does not inherently induce gradient instability in simplified GCNs. Rather, it scales the gradients according to the graph's topology and degree distribution. While this does not prove anything concerning practical scenario when using ReLU activation and BatchNorm, it shows that the graph operator does not solely worsen the gradient backpropagation in GCNs.

3.2 Oversmoothing & Oversquashing

Oversmoothing (NT and Maehara, 2019; Cai and Wang, 2020; Rusch et al., 2023) and oversquashing (Alon and Yahav, 2021; Topping et al., 2021; Giovanni et al., 2023) are two critical limitations identified in deep GNNs. The former occurs when node embeddings become indistinguishably similar with increasing network depth, thus characterized by the convergence of $\left\|X_u^{(L)} - X_v^{(L)}\right\|$ to 0 for all node pairs u, v. The latter arises when the final node representations exhibit negligible dependence on the original node features, a behavior captured by the limit $\left\|\partial X_u^{(L)} / \partial X_v\right\| \to 0$ for all u, v. Building upon our simplified formulation GCNs, we provide a unified explanation for both oversmoothing and oversquashing in this framework by drawing a parallel to spectral and Markovian convergence. Throughout this subsection, we define $\lambda_* = \max\{|\lambda| \mid \lambda \in \operatorname{Spec}(\hat{A}) \cap (-1, 1)\}$, where $\operatorname{Spec}(\hat{A})$ denotes the spectrum of the normalized adjacency matrix $\hat{A}$. Proposition 3.2 and lemma 3.0.1 directly follow from well-known results concerning the spectrum of the normalized adjacency matrix (Chung, 1997).

Proposition 3.2. *Let $\hat{A}$ be the normalized adjacency matrix of an undirected connected graph G with added self-loops:*

$$\hat{A}^L \quad = \quad I_1 + \sum_{\substack{\lambda \in \operatorname{Spec}(\hat{A}) \\ \lambda \in (-1,1)}} \lambda^L I_\lambda \quad = \quad \frac{1}{2m+n} \tilde{D}^{\frac{1}{2}} J \tilde{D}^{\frac{1}{2}} + \mathcal{O}\left(\lambda_*^L\right)$$

where I_λ is the orthogonal projection on the eigenspace associated with eigenvalue λ and $J \in \mathbb{R}^{n \times n}$ is the matrix with all entries equal to 1.

Lemma 3.0.1. *Assuming there exists $C > 0$ such that $\|W_{1 \to l}\| \leq C$ for all $l > 0$, the node-wise feature representation of a simplified GCN after the L-th layer becomes:*

$$X_u^{(L)} = \frac{\sqrt{d_u}}{2m+n} \sum_{v \in} \sqrt{d_v} X_v W_{1 \to L} + \mathcal{O}(\lambda_*^L) \quad \textit{for all } u \in 1, \dots, n$$

Using these preliminary results we derive the following theorem unifying over-squashing and over-smoothing in simplified deep GCNs as two consequences of the spectral convergence of $\hat{A}^L$:

Theorem 3.1. *Under Lemma 3.0.1's assumptions:*

$$\left\| X_u^{(L)} - X_v^{(L)} \right\| = \frac{|\sqrt{d_u} - \sqrt{d_v}|}{2m+n} \cdot \left\| \sum_v \sqrt{d_v} X_v W_{1\to L} \right\| + \mathcal{O}(\lambda_*^L) \tag{1}$$

and

$$\frac{\partial X_u^{(L)}}{\partial X_v} = \frac{\sqrt{d_u d_v}}{2m+n} W_{1\to L}^\top + \mathcal{O}(\lambda_*^L). \tag{2}$$

Although lemma 3.0.1 and equation 1 of theorem 3.1 respectively demonstrate that rank-one collapse and output smoothness (two standard characterizations of oversmoothing) inevitably arise in simplified GCNs as the depth increases, equation 2 of theorem 3.1 offers a refined perspective on oversquashing. In line with the counterexamples provided by Arnaiz-Rodriguez and Errica (2025) and using classical links between spectral gap and topological bottlenecks (Chung, 1997), it reveals that the connection between asymptotical output-input insensitivity and the presence of topological bottlenecks lies solely in the convergence rate. Notably, the graph's influence on asymptotic sensitivity is captured exclusively through the ratio $\sqrt{d_u d_v}/(2m+n)$. For instance, two adjacent nodes u and v located in the interior of a large 2D grid graph satisfy $\partial X_u^{(L)}/\partial X_v \approx 4/(2m+n) W_{1\to L} \approx 0$ (under lemma 3.0.1's assumptions) for some suitably large L, despite the absence of any topological bottleneck. A more detailed discussion on the conditions for reaching this asymptotic regime in practical scenarios is provided in Appendix 8.

Interestingly, this spectral behavior is not specific to $\hat{A}$ but is in fact characteristic of a broad class of stochastic matrices. Motivated by this observation, we now adopt a Markovian perspective, which provides a more general and principled framework to understand these limitations. This broader framework allows us to characterize the behavior of a larger class of MPNNs. We begin by noting that the spectral properties of $\hat{A}$ are not coincidental: they arise naturally when interpreting $\hat{A}^L$ in terms of Markov chains. Specifically, $\hat{A}^L$ can be written as $\tilde{D}^{1/2} P^L \tilde{D}^{-1/2}$, where $P = \tilde{D}^{-1}\tilde{A}$ is a row-stochastic matrix. This leads directly to proposition 3.3 which follows as corollary of a well known result:

Proposition 3.3 (Markovian convergence in deep GCNs)**.** *Let G be connected and undirected with self-loops. Then the random walk matrix $P = \tilde{D}^{-1}\tilde{A}$ has a unique stationary distribution $\pi \in \mathbb{R}^n$ given by $\pi_i = \frac{d_i+1}{2m+n}$. Moreover,*

$$\hat{A}^L X W_{1\to L} = \Pi X W_{1\to L} + \mathcal{O}(\lambda_*^L)$$

where $\Pi := \tilde{D}^{1/2} \mathbf{1}_n \pi^\top \tilde{D}^{-1/2}$ is a rank-one projection.

This result connects our setting to the well-established theory of Markov chains. In particular, Hajnal and Bartlett (1958) extend this connection beyond fixed transition matrices. Motivated by this, we introduce a broader class of simplified GNNs (e.g., GAT) to generalize theorem 3.1.

Definition 3.1 (Markovian-MPNNs)**.** *We call* Markovian-MPNN *any sequence $(f_l)_{l\in 1,\ldots}$ of simplified GNN layers such that:*

$$f_l(X) = P^{(l)} X W^{(l)},$$

where $W^{(l)}$ is a learnable weight matrix and $P^{(l)}$ is a stochastic matrix.

This definition encompasses any simplified layer where aggregation is implemented as a convex combination of node features across the graph. As the product of stochastic matrices remains stochastic, the following lemma extends the vanishing gradient analysis to the case of *Markovian-MPNNs*.

Proposition 3.4. *Let $\alpha \in \mathbb{R}^{n\times n}$ be a row-stochastic matrix. Then:*

$$1 \leq \mathrm{Tr}(\alpha^\top \alpha) \leq n.$$

Moreover, these bounds are tight: The lower bound is attained when each row of α is the uniform distribution, i.e., $\alpha_{i,:} = \frac{1}{n}\mathbf{1}_n$ for all $i \in 1, \ldots, n$. The upper bound is attained when each row is a one-hot vector, i.e., for each i, there exists a unique j such that $\alpha_{i,j} = 1$ and $\alpha_{i,k} = 0$ for $k \neq j$.

To conclude, we present a direct application of Lemma 3 from Hajnal and Bartlett (1958), which constitutes the main result of this section. For any matrix $Q \in \mathbb{R}^{n\times n}$, we define its overlap coefficient as $\gamma(Q) := \min_{u\neq v} \sum_{w=1}^n \min(Q_{u,w}, Q_{v,w})$.

Theorem 3.2 (Convergence of Markovian-MPNNs)**.** *Let* $(f_l)_{l \in 1,...}$ *be a* Markovian-MPNN*. Then,*

$$\left\| X_u^{(l)} - X_v^{(l)} \right\|_\infty \leq (1-\varepsilon)^{\alpha(l)} \left\| X W_{1\to l} \right\|_1 \quad \textit{for all pair of nodes } u, v$$

where $\alpha(l) + 1$ *is the maximal length of a sequence* $1 < i_1, \ldots, i_k = l$ *such that* $\gamma\left(P^{(i_{\beta+1})} \cdots P^{(i_\beta)}\right) \geq \varepsilon > 0$ *for all* $\beta \in 1, \ldots, k-1$*. Moreover, Assuming there exists* $C > 0$ *such that* $\|W_{1\to l}\| \leq C$ *for all* $l > 0$*, there exists a probability distribution* π *(independent of* u*) such that:*

$$\frac{\partial X_u^{(l)}}{\partial X_v} = \pi_v W_{1\to l}^\top + \mathcal{O}\left((1-\varepsilon)^{\alpha(l)}\right)$$

Furthermore, stronger results can be derived for specific architectures. For instance, in attention-based networks, where attention weights are recomputed from updated features at each layer, a snowball effect emerges: as the rank of node features collapses, so does the rank of the resulting stochastic aggregators, exponentially compounding expressivity loss (Dong et al., 2023). Since fundamental limitations of Markovian MPNNs appear to arise from the spectral properties of their stochastic aggregation matrices, we turn to alternative GNN architectures that retain the simplicity of the original formulation while replacing the aggregation operator with unitary matrices. Although unitary matrices have been successfully employed in RNNs (Arjovsky et al., 2016; Jing et al., 2017; Dees et al., 2019), their application as graph aggregation operators necessitates compatibility with the underlying graph structure. Thus, in the following section, we investigate operators inspired by Quantum Walks as topology-aware unitary alternatives.

4 Quantum vs classic dynamics

4.1 Continuous Time Quantum Walks

Theorems 3.1 and 3.2 suggest that the inability of classical message-passing GNNs to capture interactions very distant nodes stem from the convergence properties of their associated Markovian dynamics. Repeated applications of stochastic aggregation operators lead to the loss of discriminative information across distant nodes, effectively collapsing features toward a rank-one projection. In constrast, Continuous-Time Quantum Walks (CTQWs) introduced by Farhi and Gutmann (1998), evolve according to unitary dynamics governed by the Schrödinger equation:

$$\frac{d}{dt}\psi(t) = -iL\psi(t), \quad \psi(t) = e^{-itL}\psi(0),$$

where L is the combinatorial graph Laplacian and $\psi(t) \in \mathbb{C}^n$ is the quantum state. Contrary to the classical diffusion equation:

$$\frac{d}{dt}x(t) = -Lx(t), \quad x(t) = e^{-tL}x(0),$$

which solutions fall in the *Markovian* processes category. In CTQWs, the unitarity of the evolution operator $U(t) = e^{-itL}$, preserves both the euclidean norm of the signal and the gradients.

If $L = \sum_\lambda \lambda I_\lambda$, then:

$$\psi(t) = \sum_\lambda e^{-it\lambda} I_\lambda \psi(0),$$

i.e., each eigenmode rotates in phase rather than decaying (as in $e^{-t\lambda}$ for diffusion). This preserves amplitude and allows information to propagate across the graph without energy loss. Continuous-time quantum walks (CTQWs) have been extensively studied in the contexts of quantum computation and mathematical physics. However, their application within Graph Neural Networks remains relatively underexplored. While prior works by Dernbach et al. (2019) and Yu et al. (2024) have investigated discrete-time quantum walks or coin-based formulations, we instead adopt a continuous-time formulation by using the operator $U = e^{-iL}$ as a feature propagation kernel. We argue that this approach provides a more principled framework — one that is both easier to implement and more naturally comparable to classical Markovian dynamics. To our knowledge, Kiani et al. (2024) explores the only analoguous approach by using the adjacency matrix as a Hamiltonian. To better understand how CTQW dynamics differ from their classical counterparts, we now compare them against the heat kernel, a well-established model of diffusion on graphs.

4.2 Heat Kernel as a Benchmark for Continuous-Time Quantum Walks

To highlight the fundamental differences between classical and quantum-inspired propagation on graphs, we compare their long-term behavior and exploration dynamics. Classical diffusion, modeled via the heat kernel e^{-tL}, induces a dissipative process that converges to a uniform stationary distribution. In contrast, CTQWs are governed by the unitary operator e^{-itL}, precluding convergence in the classical sense. To enable comparison, we examine the time-averaged occupation measures:

$$p_{u\to v}(T) := \frac{1}{T}\int_0^T [e^{-tL}]_{u,v}\,dt \quad \text{and} \quad q_{u\to v}(T) := \frac{1}{T}\int_0^T \left|[e^{-itL}]_{u,v}\right|^2 dt$$

Using the spectral decomposition of the combinatorial graph Laplacian (Chung, 1997)

$$e^{-tL} = \frac{1}{n}J + \sum_{\lambda>0} e^{-t\lambda} I_\lambda$$

the heat kernel average expands as:

$$p_{u\to v}(T) = \frac{1}{n} + \sum_{\lambda>0} I_\lambda(u,v)\cdot\frac{1-e^{-T\lambda}}{T\lambda}$$

As $T \to \infty$, the spectral components vanishes and $p_{u\to v}(T)$ converges to $\frac{1}{n}$, regardless of the starting node u. This convergence reflects the behavior of Markovian-MPNNs, making the heat kernel a principled benchmark to analyze markovian models and compare them to CTQWs.

By contrast, the CTQW time average reads:

$$q_{u\to v}(T) = \sum_\lambda [I_\lambda(u,v)]^2 + 2\sum_{\lambda<\lambda'} I_\lambda(u,v)I_{\lambda'}(u,v)\cdot\frac{\sin(T(\lambda'-\lambda))}{T(\lambda'-\lambda)}$$

This formulation reveals key distinctions:

- **Non-uniform limiting behavior:** As $T \to \infty$, the interference terms vanish, yielding

$$q_{u\to v}(\infty) = \sum_\lambda [I_\lambda(u,v)]^2,$$

  which retains dependence on both u and v, encoding long-term memory of the initial state.

- **Spectral interference:** The oscillating behavior is governed by interactions between spectral components $I_\lambda(u,v)I_{\lambda'}(u,v)$ and the eigenvalue differences $\lambda' - \lambda$, in contrast to the independent exponential decay in classical diffusion.

This analysis shows that CTQWs maintain richer structural dependencies and avoid dissipative homogenization. This makes them a promising alternative to classical diffusion in GNNs, particularly in tasks where long-range interactions and preservation of node-specific information are critical.

5 An interpretable learned bias

CTQWs and heat diffusion dynamics are highly sensitive to the choice of generator. The combinatorial Laplacian assumes uniform treatment of nodes and edges, leading to isotropic diffusion. To enable adaptive control over quantum propagation, we introduce a biased Laplacian operator L_μ, which incorporates a learned bias $\mu : V \to \mathbb{R}_{\geq 0}$. This bias modifies the effective geometry of the graph, steering the quantum evolution and the heat diffusion in a task-adaptive manner — without altering the underlying topology or connectivity.

5.1 Definition

We define L_μ through its associated Dirichlet form:

$$g^\top L_\mu f = \frac{1}{2}\sum_{u\in V}\mu(u)\sum_{v\sim u}(g(v)-g(u))(f(v)-f(u)),$$

which symmetrically accounts for local variation weighted by μ. This leads to the operator formulation:

$$L_\mu = D_\mu - A_\mu,$$

where the entries of $A_\mu \in \mathbb{R}^{n\times n}$ are given by

$$[A_\mu]_{ij} = \begin{cases} \frac{1}{2}(\mu(i)+\mu(j)) & \text{if } i \sim j, \\ 0 & \text{otherwise,} \end{cases}$$

and D_μ is the corresponding diagonal degree matrix, with $[D_\mu]_{ii} = \sum_{j\sim i}[A_\mu]_{ij}$. Although this results in a weighted Laplacian with edge weights of the form $w_{ij} = \frac{1}{2}(\mu(i)+\mu(j))$, this not equivalent to a general edge-weighting scheme. The critical difference lies in the parametrization: the edge weights are symmetrically derived from μ, rather than being independently learned per edge. This structure imposes a constraint that reduces expressiveness compared to fully flexible edge-weighting, but offers a meaningful trade-off. By limiting the parameter space, the model avoids overparameterization, leading to improved scalability and interpretability. Indeed, the learned bias μ carries a natural semantic interpretation as an intrinsic node-level property, which can facilitate downstream analysis or transfer across related tasks.

5.2 Implementation

The bias function μ is learned end-to-end within the neural network. Specifically, we use a single GCN layer followed by a Softplus activation to ensure smooth non-negativity:

$$\mu = \text{Softplus}(\text{GCNConv}(X)). \tag{3}$$

This small neural network is a submodule of both the heat kernel-based and the CTQW-based model, with their forward passes beginning with the computation of equation 3. Once μ is obtained, we define a biased Laplacian L_μ governing the heat diffusion via e^{-L_μ} or the quantum propagation through the unitary operator e^{-iL_μ}. Each subsequent layer of the models applies this propagation as:

$$X' = \text{BN}\left(\text{ReLU}\left(e^{-iL_\mu}XW\right)\right) \quad \text{or} \quad X' = \text{BN}\left(\text{ReLU}\left(e^{-L_\mu}XW\right)\right)$$

where W is a classic linear layer and BN denotes batch normalization. Hence, μ is computed only once at the start of the forward pass, avoiding the compounded smoothing effects that typically affect attention weights across layers. Finally, since L_μ is a real symmetric $n \times n$ matrix, both e^{-iL_μ} and e^{-L_μ} can be computed in $\mathcal{O}(n^3)$ time. This matches the single-layer theoretical complexity in the classical setting $d \leq n$. Efficient approximation methods for these operators are presented in the second part of this work.

5.3 Theoretical results

Having established the central role of the spectral decomposition of the Laplacian in governing both heat diffusion and CTQW dynamics, we now turn our attention to the biased Laplacian operator. In this section, we present key spectral properties showing how μ can be leveraged to adapt and control the propagation dynamics on the graph. To this end, we relate the spectra of L and L_μ through their Rayleigh quotients:

$$\mathcal{R}(f) = \frac{f^\top L f}{f^\top f} \quad \text{and} \quad \mathcal{R}_\mu(f) = \frac{f^\top L_\mu f}{f^\top f}$$

Definition 5.1. *Let $f \in \mathbb{R}^n$, we denote the squared euclidean norm of its local variation:*

$$N(f) = \left(\frac{\|\nabla f(u)\|_2^2}{f^\top f}\right)_{u\in V} \in \mathbb{R}^n \quad \textit{where} \quad \|\nabla f(u)\|_2^2 = \sum_{v\sim u}(f(u)-f(v))^2$$

and the associated probability distribution

$$p_f = \frac{1}{1^\top N(f)} N(f)$$

Lemma 5.0.1. *Let $f \in \mathbb{R}^n$, we have*

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

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

This factorization isolates three key contributors to the spectrum of the modulated Laplacian L_μ:

- Global scaling by $\|\mu\|_1$,
- Local correlation between μ and p_f via the μ-weighted expectation $\mathbb{E}_\mu[p_f]$,
- Spectral structure of the combinatorial Laplacian via $\mathcal{R}(f)$.

From this decomposition, we derive:

Theorem 5.1. *Let $\lambda_0 = 0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$ be the eigenvalues of L, and let λ_k^μ denote the eigenvalues of L_μ, ordered similarly. Let $f_0, \ldots, f_{n-1}$ and $g_0, \ldots, g_{n-1}$ be two orthonormal bases of eigenvectors associated with L. For all $k \in 0, \ldots, n-1$:*

$$\lambda_k \; \|\mu\|_1 \min_{\substack{g \in Span(g_k, \ldots, g_{n-1}) \\ \|g\|_2 = 1}} \mathbb{E}_\mu[p_g] \leq \lambda_k^\mu \leq \lambda_k \; \|\mu\|_1 \max_{\substack{f \in Span(f_1, \ldots, f_k) \\ \|f\|_2 = 1}} \mathbb{E}_\mu[p_f]$$

Hence, since eigenvectors associated with low (resp. high) eigenvalues tend to concentrate their variations in sparse (resp. dense) regions of the graph, this formulation enables targeted spectral modulation by appropriately shaping μ, without relying on its global scale $\|\mu\|_1$. As corollaries, we show how to modulate the spectrum and the input-output sensitivity between a given pair of nodes on the star graph, only by tuning μ. This graph is interesting as it is a common example of bottleneck or articulating point between clusters. We further reefer to $\frac{\mu}{\|\mu\|_1}$ as $\tilde{\mu}$.

Corollary 5.1.1. *Let μ be such that $\tilde{\mu}(1) = \tilde{\mu}(2) = 0$ and $\tilde{\mu}(k) = \frac{1}{n-2}$ for all $k \notin 1, 2$, then:*

$$\lambda_1^\mu \leq \frac{1}{2(n-2)} \|\mu\|_1 \, \lambda_1$$

Corollary 5.1.2. *Let μ be such that*

$$\tilde{\mu}(u) = \begin{cases} \frac{1}{2} \text{ if } u = 0 \\ 0 \text{ if } u \in 1, 2 \\ \frac{1}{2(n-1)} + \frac{1}{(n-1)(n-3)} \text{ else} \end{cases}$$

and $\|\mu\|_1 = 2$, then simultaneously:

$$\frac{3}{2}\lambda_{n-1} \leq \lambda_{n-1}^\mu \quad \text{and} \quad \lambda_1^\mu \leq \frac{1}{2}\lambda_1$$

Proposition 5.1. *For all pair of distinct leaves $u, v \in 1, \ldots, n-1$ and for all $t \in \mathbb{R}$:*

$$|\langle u \mid e^{-itL} \mid v \rangle|^2 = \mathcal{O}\left(\frac{1}{n^2}\right)$$

Corollary 5.1.3. *For all $n \geq 4$, for all pair of distinct leaves u and v, there exists μ such that*

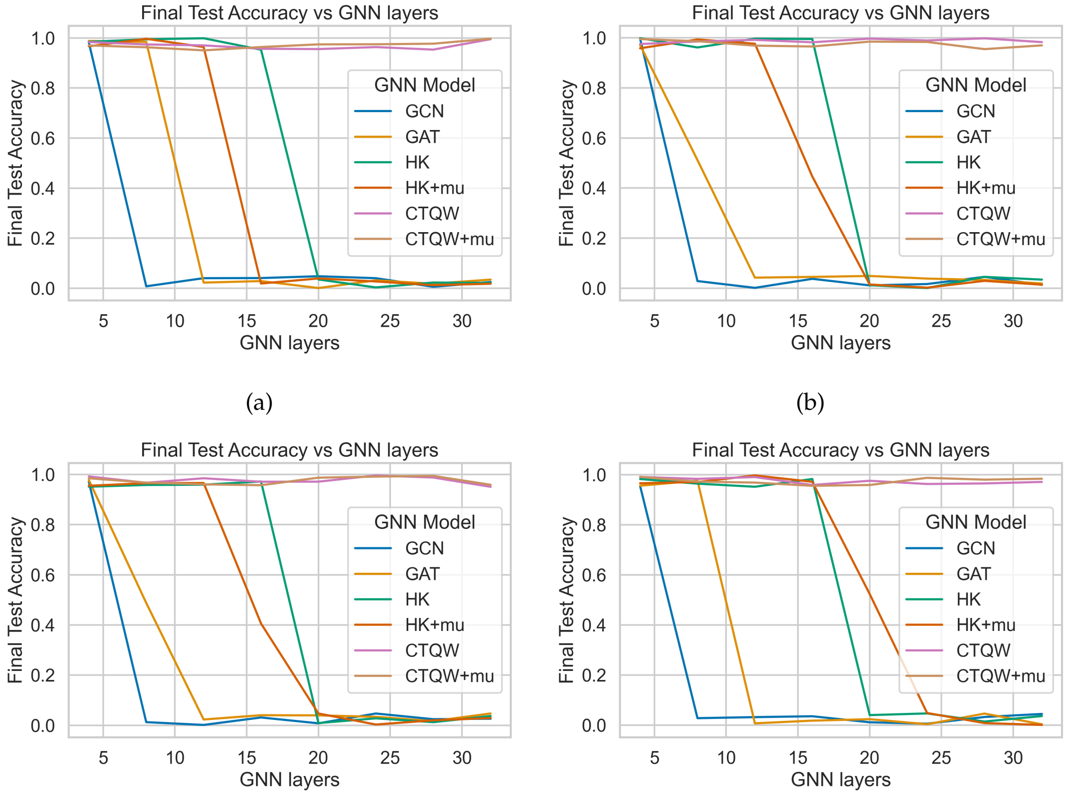
$$|\langle u \mid e^{-iL_\mu} \mid v \rangle|^2 \geq \frac{1}{4}$$

Proofs are detailed in appendix 8.

6 Experiments

To evaluate long-range propagation, we introduce four synthetic tasks. Each task consist 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. All other features are i.i.d. standard Gaussian noise. The number of GNN layers is exactly the distance between the query and the answer nodes. Each task uses a different underlying topology: Path Graph, Barbell Graph, Barbell-Star Graph, and Ring Graph. In the Ring task, one of the two paths from query to answer has all-zero features, testing whether adaptive models can effectively route information. Additionnal details and illustrating figures are available in Appendix 8.

We benchmark GCN and GAT against Heat kernel, Heat kernel + μ, CTQW and CTQW+ μ. The results in Figure 4 demonstrate that CTQW and CTQW+μ consistently outperform all other architectures. Moreover, although their theoretical computational costs are much higher than those of classical message-passing models, their learning processes converge much faster potentially making them cheaper to train in practice. Furthermore, as demonstrated in Appendix 8, μ can effectively offer valuable insights on optimal routing for biased models. Extending these insights to real-world datasets with more complex signal patterns remains a key direction for future work.



(a)

(b)

(c)

(d)

Figure 1: Test accuracy across synthetic tasks as a function of the number of layers. (a) Path Graph, (b) Barbell Graph, (c) Barbell-Star Graph, and (d) Ring Graph. In all cases, CTQW and CTQW+μ consistently outperform all other architectures.

We provide additional details and figures in appendix C.

7 A Computationally Efficient Unitary Operator

In the first part of this work, we investigated the long-range limitations of simplified GNNs from both spectral and dynamical perspectives. To address these limitations, we introduced Continuous-Time Quantum Walks (CTQWs) as a novel propagation mechanism. To control the dynamics of CTQWs and heat diffusion, we proposed the biased Laplacian L_μ, which incorporates a learnable node-level bias while preserving the underlying graph topology. This design enables dynamic modulation of the propagation behavior and enhances model interpretability. Through synthetic tasks specifically designed to test long-range reasoning, we demonstrated that CTQW-based models consistently outperform both classical GNNs and standard diffusion-based methods, in terms of accuracy and convergence speed. Furthermore, we showed that the biased heat kernel benefits from a single learned aggregation operator, outperforming other non-quantum baselines as predicted by our long-range analysis. However, although CTQW-based models converge faster, their per-layer computational cost remains significant due to the reliance on matrix spectral decomposition.

In this section, we introduce a computationally efficient unitary graph operator leveraging Chebyshev polynomials. We retain the following propagation rule:

$$X' = \mathrm{BN}\left(\sigma(VXW)\right),$$

Here, V is a unitary operator defined as:

$$V = T_k(\hat{A}) + (I - \hat{A}^2)^{\frac{1}{2}} U_{k-1}(\hat{A})$$

where T_k (resp. U_k) is the k-th Chebyshev polynomial of the first (resp. second) kind and k is a fixed integer. As $\hat{A}$ is real symmetric and its spectrum lies in $[-1, 1]$, the above rewrites:

$$V = \sum_{\lambda \in \mathrm{Spec}(\hat{A})} \left(T_k(\lambda) + \sqrt{1 - \lambda^2}\, U_{k-1}(\lambda) \right) I_\lambda = \sum_{\lambda \in \mathrm{Spec}(\hat{A})} e^{ik \arccos(\lambda)} I_\lambda.$$

hence the unitarity of V. Using sparse tensor product, the terms $T_k(\hat{A})$ and $U_{k-1}(\hat{A})$ can be computed using $\mathcal{O}^*(km)$ additions and multiplications (where m is the number of edges) via recurrence relations. However, computing the square root $(I - \hat{A}^2)^{1/2}$ directly is costly, as it requires computing the diagonalization of $\hat{A}$ — taking $\mathcal{O}^*(n^3)$ operations, where n is the number of nodes. To address this, we approximate it via a truncated power series expansion:

$$(I - \hat{A}^2)^{1/2} \approx \sum_{i=0}^{c} \binom{c}{i} (-1)^i \hat{A}^{2i} + \mathcal{O}(\hat{A}^{2c}),$$

which provides a viable approximation except at the spectral extremities $\lambda = \pm 1$. Despite this limitation, the proposed model achieves the performances of the full CTQW model in synthetic tasks, while significantly reducing computational overhead.

8 Conclusion

Graph Neural Networks have transformed the way we approach learning from relational data by directly leveraging graph structures to capture both attribute and topological information. However, as these models grow deeper and more expressive, they confront inherent challenges such as vanishing gradients, oversmoothing, and oversquashing—phenomena that degrade performance and limit their effectiveness in capturing long-range interactions.

This work addresses these issues by unifying oversmoothing and oversquashing under a common framework. In the first phase of the internship, we proposed two novel architectures, grounded in quantum dynamics and biased diffusion, to mitigate these effects within the Markovian-MPNN framework. In the second phase, we refined these approaches through the development of a Chebyshev-based model inspired by ChebNet, offering a promising balance between expressivity and computational efficiency.

Together, these contributions deepen our understanding of the limitations of deep GNNs and provide new architectural tools to overcome them. As GNNs continue to expand into more complex domains—including dynamic graphs, higher-order structures, and generative models—addressing these foundational bottlenecks will be crucial to unlocking their full potential.

References

U. Alon and E. Yahav. On the bottleneck of graph neural networks and its practical implications. In *ICLR*, 2021.

M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks, 2016. URL https://arxiv.org/abs/1511.06464.

A. Arnaiz-Rodriguez and F. Errica. Oversmoothing, ”oversquashing”, heterophily, long-range, and more: Demystifying common beliefs in graph machine learning, 2025. URL https://arxiv.org/abs/2505.15547.

Arroyo, A. Gravina, B. Gutteridge, F. Barbero, C. Gallicchio, X. Dong, M. Bronstein, and P. Vandergheynst. On vanishing gradients, over-smoothing, and over-squashing in gnns: Bridging recurrent and graph learning. *arXiv preprint arXiv:2502.10818*, 2025.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2014.

C. Cai and Y. Wang. A note on over-smoothing for graph neural networks, 2020. URL https://arxiv.org/abs/2006.13318.

F. Chung and L. Lu. The diameter of sparse random graphs. *Advances in Applied Mathematics*, 26(4): 257–279, 2001. ISSN 0196-8858. doi: https://doi.org/10.1006/aama.2001.0720. URL https://www.sciencedirect.com/science/article/pii/S0196885801907201.

F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

B. S. Dees, L. Stankovic, M. Dakovic, A. G. Constantinides, and D. P. Mandic. Unitary shift operators on a graph, 2019. URL https://arxiv.org/abs/1909.05767.

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2017. URL https://arxiv.org/abs/1606.09375.

S. Dernbach, A. Mohseni Kabir, S. Pal, M. Gepner, and D. Towsley. Quantum walk neural networks with feature dependent coins. *Applied Network Science*, 4:76, 09 2019. doi: 10.1007/s41109-019-0188-2.

Y. Dong, J.-B. Cordonnier, and A. Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth, 2023. URL https://arxiv.org/abs/2103.03404.

E. Farhi and S. Gutmann. Quantum computation and decision trees. *Physical Review A*, 58(2):915–928, Aug. 1998. ISSN 1094-1622. doi: 10.1103/physreva.58.915. URL http://dx.doi.org/10.1103/PhysRevA.58.915.

J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

F. D. Giovanni, L. Giusti, F. Barbero, G. Luise, P. Lio’, and M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology, 2023. URL https://arxiv.org/abs/2302.02941.

J. Hajnal and M. S. Bartlett. Weak ergodicity in non-homogeneous markov chains. *Mathematical Proceedings of the Cambridge Philosophical Society*, 54(2):233–246, 1958. doi: 10.1017/S0305004100033399.

W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pages 1025–1035, 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

C. Hoffman, M. Kahle, and E. Paquette. Spectral gaps of random graphs and applications. *International Mathematics Research Notices*, 2021(11):8353–8404, May 2019. ISSN 1687-0247. doi: 10.1093/imrn/rnz077. URL http://dx.doi.org/10.1093/imrn/rnz077.

L. Jing, Y. Shen, T. Dubcek, J. Peurifoy, S. Skirlo, Y. LeCun, M. Tegmark, and M. Soljačić. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1733–1741. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jing17a.html.

B. T. Kiani, L. Fesser, and M. Weber. Unitary convolutions for learning on graphs and groups, 2024. URL https://arxiv.org/abs/2410.05499.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2017.

H. NT and T. Maehara. Revisiting graph neural networks: All we have is low-pass filters, 2019. URL https://arxiv.org/abs/1905.09550.

M. Park, S. Choi, J. Heo, E. Park, and D. Kim. The oversmoothing fallacy: A misguided narrative in gnn research, 2025. URL https://arxiv.org/abs/2506.04653.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks, 2013. URL https://arxiv.org/abs/1211.5063.

T. K. Rusch, M. M. Bronstein, and S. Mishra. A survey on oversmoothing in graph neural networks, 2023. URL https://arxiv.org/abs/2303.10993.

J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *ICML*, 2021.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

L. Yu, H. Chen, J. Lv, and L. Yang. Gqwformer: A quantum-based transformer for graph representation learning, 2024. URL https://arxiv.org/abs/2412.02285.

Appendix A

Proof of Proposition 3.1

Note that by similarity, $\mathrm{Tr}(\hat{A}^k) = \mathrm{Tr}((\tilde{D}^{-1}\tilde{A})^k)$ and $P = \tilde{D}^{-1}\tilde{A}$ is row-stochastic, so for $k \geq 1$

$$[P^k]_{ii} = (P^{k-1}P)_{ii} = \sum_j [P^{k-1}]_{ij} P_{ji} = \sum_{j \sim i} [P^{k-1}]_{ij} \frac{1}{d_j}.$$

Since $d_j \geq d_{\min}$ we have $\frac{1}{d_j} \leq \frac{1}{d_{\min}}$, hence

$$[P^k]_{ii} \leq \frac{1}{d_{\min}} \sum_j [P^{k-1}]_{ij} = \frac{1}{d_{\min}},$$

because each row of P^{k-1} sums to 1. The same reasonning yields the lower bound.

Discussion on practical scenarios

While Lemma 3.0.1 and Theorem 3.1 describe the asymptotic behavior of simplified GCNs in the limit $L \to \infty$, in practice the depth L of deep GCNs typically remains bounded by an estimate of the graph's diameter. It is therefore important to understand the conditions under which the asymptotic regime is reached quickly enough to be relevant at such realistic depths.

To estimate the graph densities required for the asymptotic regime to emerge at realistic depths, we refer to the results of (Hoffman et al., 2019), who analyze spectral properties of Erdős–Rényi random graphs. They show that if the edge probability satisfies

$$p \geq \left(\frac{1}{2} + \delta\right) \frac{\log(n)}{n}$$

for some constant $\delta > 0$, then with high probability:

$$\lambda_* \leq \frac{C}{\sqrt{p(n-1)}}$$

for some constant C.

This implies that even in relatively sparse graphs with $p = \frac{\log(n)}{n}$, the spectral gap is sufficiently large to ensure rapid convergence. Choosing the number of layers $L = 2 * \frac{\log(n)}{\log(\log(n))}$ as an estimate of the graph's diameter (Chung and Lu, 2001), the contribution of nonzero-frequency components decays as

$$\lambda_*^L \leq \left(C\sqrt{\frac{n}{\log(n)(n-1)}} \right)^{2\frac{\log(n)}{\log(\log(n))}}.$$

This decay is illustrated in 2, showing that the asymptotic behavior we characterize is effectively reached even for relatively sparse graphs of moderated size.

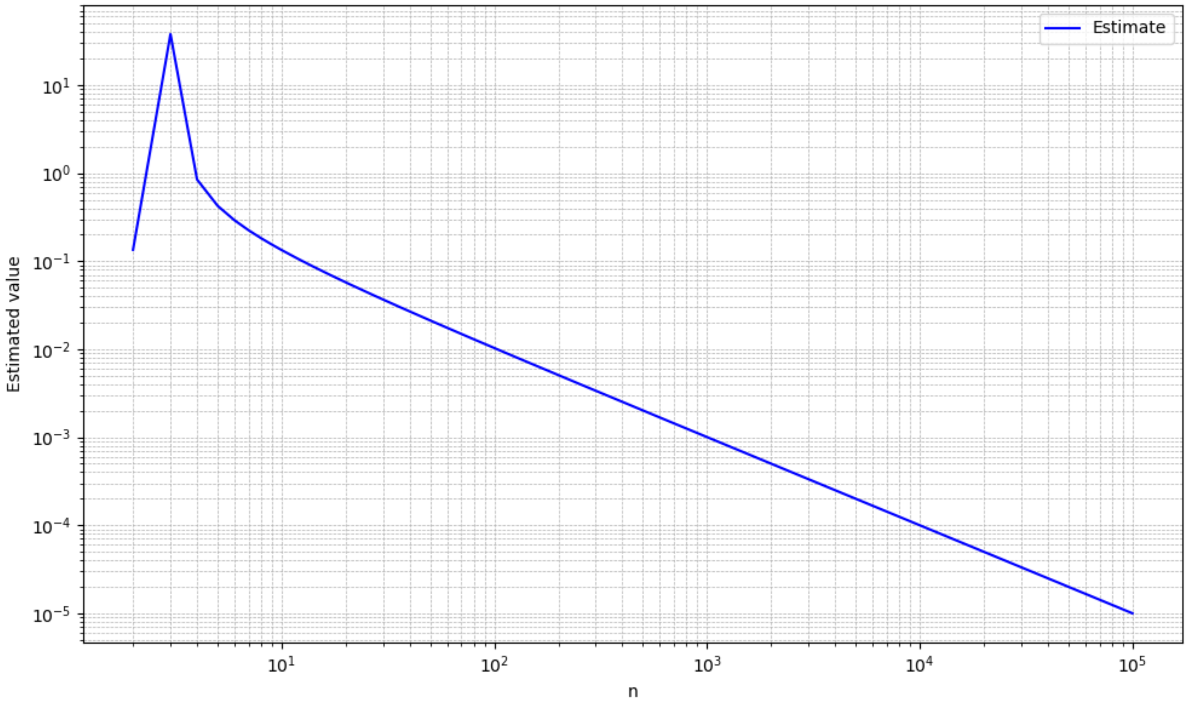


Figure 2: Estimate of the upper bound on λ_*^L with respect to n, taking $C = 1$.

Proof of Theorem 3.2

Proof. Let's fix $u \neq v \in V$ and a sequence $1 < i_1, \dots, i_k = l$ of length $\alpha(l) + 1$ such that $\gamma\left(P^{(i_{\beta+1})} \cdots P^{(i_\beta)}\right) \geq \varepsilon > 0$ for all $\beta \in 1, \dots, k-1$. Then:

$$\begin{aligned}
\left\| X_u^{(l)} - X_v^{(l)} \right\|_\infty &= \max_{k \in 1,\dots,d} |(\delta_u - \delta_v) P_{l \leftarrow 1} X W_{1 \to l} \delta_k| \\
&= \max_{k \in 1,\dots,d} |\sum_{w \in V} (\delta_u - \delta_v) P_{l \leftarrow 1} \delta_w \delta_w X W_{1 \to l} \delta_k| \\
&\leq \max_{k \in 1,\dots,d} \sum_{w \in V} |(\delta_u - \delta_v) P_{l \leftarrow 1} \delta_w| |\delta_w X W_{1 \to l} \delta_k| \quad (4) \\
&\leq \left(\max_{w \in V} |(\delta_u - \delta_v) P_{l \leftarrow 1} \delta_w| \right) \|X W_{1 \to l}\|_1 \quad (5) \\
&\leq \left(\max_{u \neq v} \max_{w \in V} |(\delta_u - \delta_v) P_{l \leftarrow 1} \delta_w| \right) \|X W_{1 \to l}\|_1 \quad (6) \\
&\leq \prod_{\beta=1}^{k-1} \left(1 - \gamma(P^{(i_{\beta+1})} \cdots P^{(i_\beta)}) \right) \|X W_{1 \to l}\|_1 \quad (7) \\
&\leq (1-\varepsilon)^{\alpha(l)} \|X W_{1 \to l}\|_1
\end{aligned}$$

Inequality (7) follows from Lemma 3 of (Hajnal and Bartlett, 1958).

Discussion. The bound above can be quite loose in practice:

The triangle inequality in (4) neglects potential cancellation effects due to positive and negative feature values. The use of the ℓ_1-norm in (5) can overestimate the actual magnitude, especially if representations are evenly distributed across features — potentially incurring a factor d overhead. Finally, the maximum taken over all pairs of distinct nodes in (6) may severely overbound the specific difference between two strongly connected neighbors u and v.

□

Appendix B

Proof of Lemma 5.0.1

Proof. We simply write:

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

□

Proof of Theorem 5.1

Proof. We only prove the upper bound as the same reasonning applies to the lower bound. Let's define the set of euclidian subspaces of $\mathbb{R}^n$, $\mathcal{F}_k = \{F, \dim(F) = k+1$ and $\lambda_k = \max\{\mathcal{R}(f) \mid f \in F, ||f||_2 = 1\}\}$ and fix $F \in \mathcal{F}_k$. By lemma 5.0.1

$$\begin{aligned}\max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}_\mu(f) &= \max_{\substack{f\in F\\ ||f||_2=1}} \|\mu\|_1 \, \mathbb{E}_\mu[p_f]\mathcal{R}(f)\\ &\leq \|\mu\|_1 \left(\max_{\substack{f\in F\\ ||f||_2=1}} \mathbb{E}_\mu[p_f]\right)\left(\max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}(f)\right)\\ &= \lambda_k \, \|\mu\|_1 \max_{\substack{f\in F\\ ||f||_2=1}} \mathbb{E}_\mu[p_f]\end{aligned}$$

Thus

$$\min_{F\in\mathcal{F}_{k+1}} \max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}_\mu(f) \leq \lambda_k \, \|\mu\|_1 \min_{F\in\mathcal{F}_{k+1}} \max_{\substack{f\in F\\ ||f||_2=1}} \mathbb{E}_\mu[p_f]$$

And

$$\min_{\substack{F\subset\mathbb{R}^n\\ \dim(f)=k+1}} \max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}_\mu(f) \leq \min_{F\in\mathcal{F}_{k+1}} \max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}_\mu(f)$$

where

$$\min_{\substack{F\subset\mathbb{R}^n\\ \dim(f)=k+1}} \max_{\substack{f\in F\\ ||f||_2=1}} \mathcal{R}_\mu(f) = \lambda_k^\mu$$

by the min-max theorem.

□

Proof of Corollary 5.1.1

Proof. Let us consider λ_1^μ, the spectral gap of L_μ. The right hand side of the previous inequality reads:

$$\lambda_1^\mu \leq \lambda_1 \, \|\mu\|_1 \mathbb{E}_\mu[p_{f_1}]$$

where f_1 is an eigenvector of L associated with λ_1.

Let S_n be the star graph with one central node u_0 connected to $n-1$ leaves. The standard Laplacian L of S_n has eigenvalues:

- $\lambda_0 = 0$ (constant eigenvector),
- $\lambda_1 = 1$ (multiplicity $n-2$),
- $\lambda_{n-1} = n$ (associated to the central node's variation).

The eigenspace of λ_1 is spanned by the basis $\{f_i\}_{i\in 1,\dots,n-2}$, defined by:

$$f_i(j) = \begin{cases} \frac{1}{\sqrt{2}} \text{ if } j = 1 \\ -\frac{1}{\sqrt{2}} \text{ if } j = i+1 \\ 0 \text{ else} \end{cases}$$

These vectors all have unit norm but are not orthogonal. However, since our analysis only involves f_1 independently of the others, there is no need to modify it. Thus

$$||\nabla f_1(u)||_2^2 = \begin{cases} 1 \text{ if } u = 0 \\ \frac{1}{2} \text{ if } u = 1 \\ \frac{1}{2} \text{ if } u = 2 \\ 0 \text{ else} \end{cases}$$

and

$$p_{f_1}(u) = \begin{cases} \frac{1}{2} \text{ if } u = 0 \\ \frac{1}{4} \text{ if } u \in 1, 2 \\ 0 \text{ else} \end{cases}$$

Setting $\mu(0) = \mu(1) = \mu(2) = 0$ and $\mu(k) = \frac{1}{n-3}$ for all $k \in 3, \dots, n-1$ might appear effective for reducing the spectral gap, but it actually forces $\lambda_1^\mu = 0$, since the edges u_0u_1 and u_0u_2 are effectively removed—disconnecting the graph. It is easy to imagine how this could be undesirable, as our interest in the star graph stems from its role as a common bridge between distincts clusters in large graphs.

A safer and still effective alternative is to set only $\mu(1) = \mu(2) = 0$ and $\mu(k) = \frac{1}{n-2}$ elsewhere. Indeed, using our bound one can verify that the spectral gap of L_μ now satisfies:

$$\lambda_1^\mu \leq \frac{\lambda_1}{2(n-2)} ||\mu||_1$$

while the connectivity is preserved and the norm of μ remains unchanged. In this case, the link between nodes u_1, u_2, and the rest of the graph is weakened rather than cut, reducing the flow of heat between potential associated clusters without fully isolating them. □

Proof of Corollary 5.1.2

Proof. As the left-hand side of our inequality becomes:

$$\lambda_{n-1}\ ||\mu||_1 \mathbb{E}_\mu[p_{g_{n-1}}] \leq \lambda_{n-1}^\mu$$

where g_{n-1} is a unit-norm eigenvector of L associated with λ_{n-1}, given by:

$$g_{n-1}(u) = \begin{cases} \sqrt{\frac{n-1}{n}} \text{ if } u = 0 \\ -\frac{1}{\sqrt{n(n-1)}} \text{ else} \end{cases}$$

From this, the local variation's norm vector becomes:

$$||\nabla g_{n-1}(u)||_2^2 = \begin{cases} \frac{n-1}{n}(\sqrt{n-1} + \frac{1}{\sqrt{n-1}})^2 \text{ if } u = 0 \\ \frac{1}{n}(\sqrt{n-1} + \frac{1}{\sqrt{n-1}})^2 \text{ else} \end{cases}$$

implying:

$$p_{g_{n-1}}(u) = \begin{cases} \frac{1}{2} \text{ if } u = 0 \\ \frac{1}{2(n-1)} \text{ else} \end{cases}$$

We now design a distribution μ to simultaneously reduce the spectral gap and increase the spectral radius.

$$\mu(u) = \begin{cases} \frac{1}{2} \text{ if } u = 0 \\ 0 \text{ if } u \in 1, 2 \\ \frac{1}{2(n-1)} + \frac{1}{(n-1)(n-3)} \text{ else} \end{cases}$$

Using our lower bound:

$$\frac{3}{4}\lambda_{n-1}||\mu||_1 \leq \lambda^{\mu}_{n-1}$$

and

$$\lambda^{\mu}_1 \leq \frac{\lambda_1}{4}||\mu||_1$$

Hence, taking $||\mu||_1 = 2$ yields simultaneously

$$\frac{3}{2}\lambda_{n-1} \leq \lambda^{\mu}_{n-1}$$

and

$$\lambda^{\mu}_1 \leq \frac{\lambda_1}{2}$$

allowing us to both decrease the spectral gap and increase the spectral radius. Additionally, one can still multiply μ by a scaling factor - to modify the global diffusion rate for instance.

This example illustrates how modulating the node-wise diffusion rates via μ allows for flexible control over the spectral properties of a graph, without altering the underlying topology. The graph is never rewired, preserving its structure and connectivity. □

Proof of Proposition 5.1

Proof. The spectral decomposition of L is:

$$L = \lambda_0 I_0 + \lambda_1 I_1 + \lambda_{n-1} I_{n-1}$$

where

- $\lambda_0 = 0$ with multiplicity 1
- $\lambda_1 = 1$ with multiplicity $n-2$
- $\lambda_{n-1} = n$ with multiplicity 1

and

- $I_0 = \frac{1}{n}J$
- $I_{n-1} = \frac{1}{n(n-1)} v_{n-1} \cdot v_{n-1}^{\top}$ with $v_{n-1} = (n-1, -1, \dots, -1)$
- $I_1 = I - I_0 - I_{n-1}$

Thus

$$e^{-itL} = \frac{1}{n}J + e^{-it}I_1 + e^{-itn}I_n$$

In particular:

$$\langle u \mid e^{-itL} \mid v \rangle = \frac{1}{n(n-1)}\left(n - 1 - ne^{-it} + e^{-itn}\right)$$

We can now express the quantity of interest:

$$\begin{aligned} |\langle u \mid e^{-itL} \mid v \rangle|^2 &= \frac{1}{(n(n-1))^2}\left(n - 1 - ne^{-it} + e^{-itn}\right)\left(n - 1 - ne^{it} + e^{itn}\right) \\ &= \frac{1}{(n(n-1))^2}\left((n-1)^2 + n^2 + 1 - 2(n-1)n\cos(t) - 2n\cos(t(n-1)) + 2(n-1)\cos(tn)\right) \end{aligned}$$

Which yields the upper bound:

$$|\langle u \mid e^{-itL} \mid v\rangle|^2 \leq \frac{1}{(n(n-1))^2}\Big((n-1)^2+n^2+1+2(n-1)n+2n+2(n-1)\Big) = \mathcal{O}\left(\frac{1}{n^2}\right)$$

□

Proof of Corollary 5.1.3

Proof. Since PGST is known to occur on P_3 we are motivated by the idea that it may be possible to increase the input-output sensitivity γ on the star graph by making it *resemble* a path, while preserving the overall connectivity and topology.

Let's follow this intuition and chose:

$$\mu(u) = \begin{cases} \alpha \text{ if } u \in 0,1,2 \\ 0 \text{ else} \end{cases}$$

The spectral decomposition of L_μ is:

$$L_\mu = \lambda_0 I_0 + \lambda_- I_- + \lambda_+ I_+ + \lambda_1 I_1 + \lambda_{n-1} I_{n-1}$$

where

- $\lambda_0 = 0$ with multiplicity 1 - $\lambda_1 = \frac{\alpha}{2}$ with multiplicity $n-4$ - $\lambda_- = \alpha\frac{n-5-\sqrt{\Delta}}{n-1-\sqrt{\Delta}}$ with multiplicity 1 - $\lambda_+ = \alpha\frac{n-5+\sqrt{\Delta}}{n-1+\sqrt{\Delta}}$ with multiplicity 1 - $\lambda_{n-1} = \alpha$ with multiplicity 1

with $\Delta = (n-1)^2+16$ and

- $I_0 = \frac{1}{n}J$ - $I_- = \frac{1}{v_-^\top v_-} v_- \cdot v_-^\top$ with $v_-(u) = \begin{cases} 4(n-9-\sqrt{\Delta}) \text{ if } u = 0 \\ (n-1-\sqrt{\Delta})(n-9-\sqrt{\Delta}) \text{ if } u \in 1,2 \\ -4(n-1-\sqrt{\Delta}) \text{ else} \end{cases}$

- $I_+ = \frac{1}{v_+^\top \cdot v_+} v_+ \cdot v_+^\top$ with $v_+(u) = \begin{cases} 4(n-9+\sqrt{\Delta}) \text{ if } u = 0 \\ (n-1+\sqrt{\Delta})(n-9+\sqrt{\Delta}) \text{ if } u \in 1,2 \\ -4(n-1+\sqrt{\Delta}) \text{ else} \end{cases}$

- $\lambda_1 = I - (I_0 + I_- + I_+ + I_{n-1})$ - $\lambda_{n-1} = v_{n-1}^\top \cdot v_{n-1}$ with $v_{n-1} = \frac{1}{\sqrt{2}}(0,1,-1,0,\dots,0)$

Thus:

$$\langle 1 \mid e^{-iL_\mu} \mid 2\rangle = \frac{1}{n} + e^{-i\lambda_-}\frac{\left(n-1-\sqrt{\Delta}\right)^2\left(n-9-\sqrt{\Delta}\right)^2}{v_-^\top \cdot v_-} + e^{-i\lambda_+}\frac{\left(n-1+\sqrt{\Delta}\right)^2\left(n-9+\sqrt{\Delta}\right)^2}{v_+^\top \cdot v_+} + e^{-i\alpha}\frac{-1}{2}$$
$$- e^{-i\frac{\alpha}{2}}\left(\frac{1}{n} + \frac{\left(n-1-\sqrt{\Delta}\right)^2\left(n-9-\sqrt{\Delta}\right)^2}{v_-^\top \cdot v_-} + \frac{\left(n-1+\sqrt{\Delta}\right)^2\left(n-9+\sqrt{\Delta}\right)^2}{v_+^\top \cdot v_+} + \frac{-1}{2}\right)$$

and

$$q_{1\to 2}(\infty) = \frac{1}{n^2} + \left[\frac{\left(n-1-\sqrt{\Delta}\right)^2\left(n-9-\sqrt{\Delta}\right)^2}{v_-^\top \cdot v_-}\right]^2 + \left[\frac{\left(n-1+\sqrt{\Delta}\right)^2\left(n-9+\sqrt{\Delta}\right)^2}{v_+^\top \cdot v_+}\right]^2 + \frac{1}{4}$$
$$+ \left[\frac{1}{n} + \frac{\left(n-1-\sqrt{\Delta}\right)^2\left(n-9-\sqrt{\Delta}\right)^2}{v_-^\top \cdot v_-} + \frac{\left(n-1+\sqrt{\Delta}\right)^2\left(n-9+\sqrt{\Delta}\right)^2}{v_+^\top \cdot v_+} + \frac{-1}{2}\right]^2$$

so for all $n \geq 4$ there exists $\varepsilon_n > 0$ such that

$$q_{1\to 2}(\infty) \geq \frac{1}{4} + \varepsilon_n$$

Thus for all $n \geq 4$, there exists t_n:

$$|\langle u \mid e^{-it_n L_\mu} \mid v\rangle|^2 \geq \frac{1}{4}$$

Finally

$$|\langle u \mid e^{-iL_{\mu_n}} \mid v\rangle|^2 \geq \frac{1}{4}$$

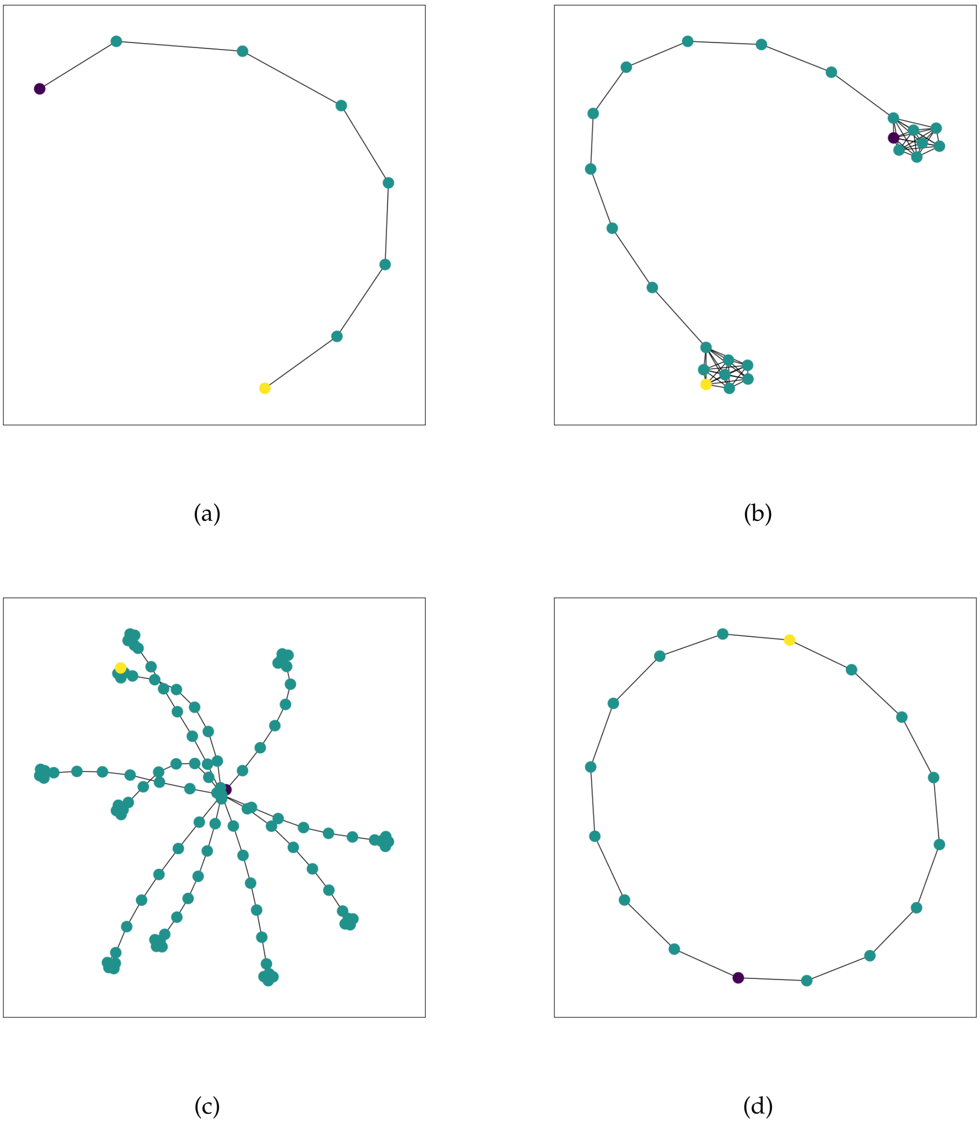
with $\mu_n = t_n \mu$. □

Appendix C

Synthetic Graph Topologies

We use the following four graph structures:

1. **Path Graph**: A simple path of n nodes, with the query node at one end and the answer node at the opposite end.
2. **Barbell Graph**: Two complete graphs (clusters) of size n, connected by a path of n nodes. The query node lies in one cluster, and the answer node in the other.
3. **Barbell-Star Graph**: A central cluster of size n is connected to 10 leaf clusters (each also of size n) by disjoint paths of length n. The query node lies in the central cluster, and the answer node is in a uniformly chosen leaf cluster.
4. **Ring Graph**: A cycle of length $2n$, with the query node at position 1 and the answer node at position n, located directly opposite in the ring. This graph admits two disjoint paths from query to answer. One path is filled with Gaussian-noise features, while the other has zero features.



(a)

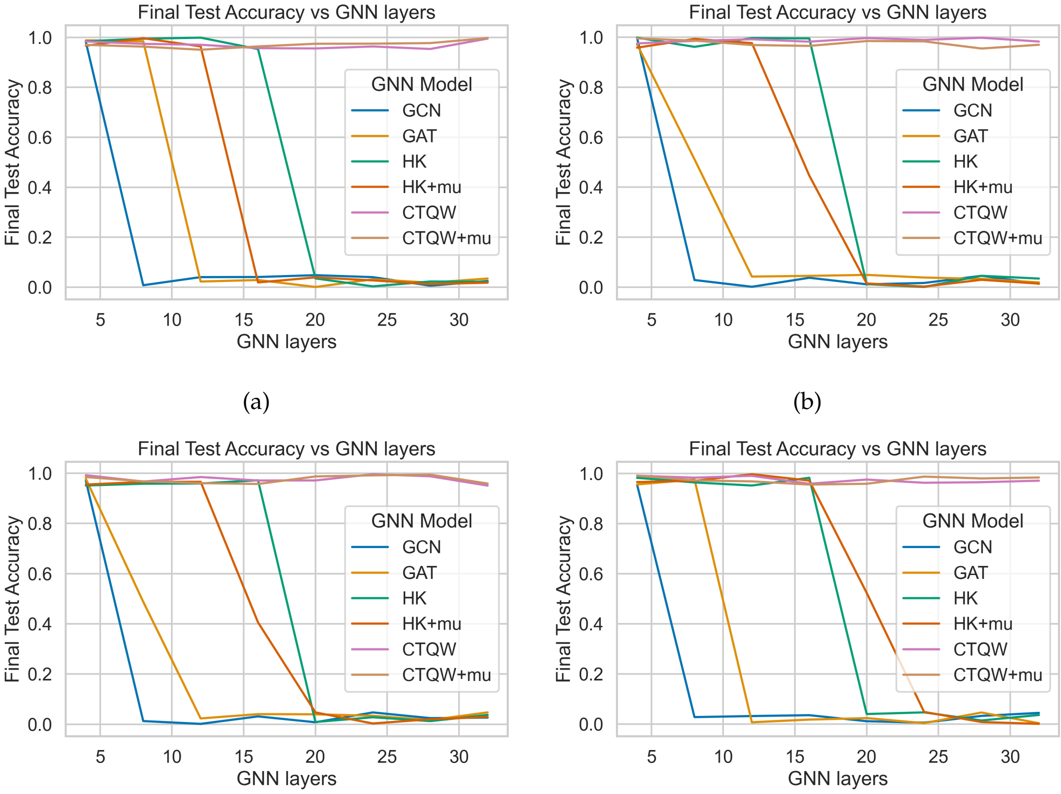
(b)

(c)

(d)

Figure 3: Graph topologies used in the synthetic tasks. (a) Path Graph, (b) Barbell Graph, (c) Barbell-Star Graph, and (d) Ring Graph. In each case, the query node is shown in purple and the answer node in yellow.

Results



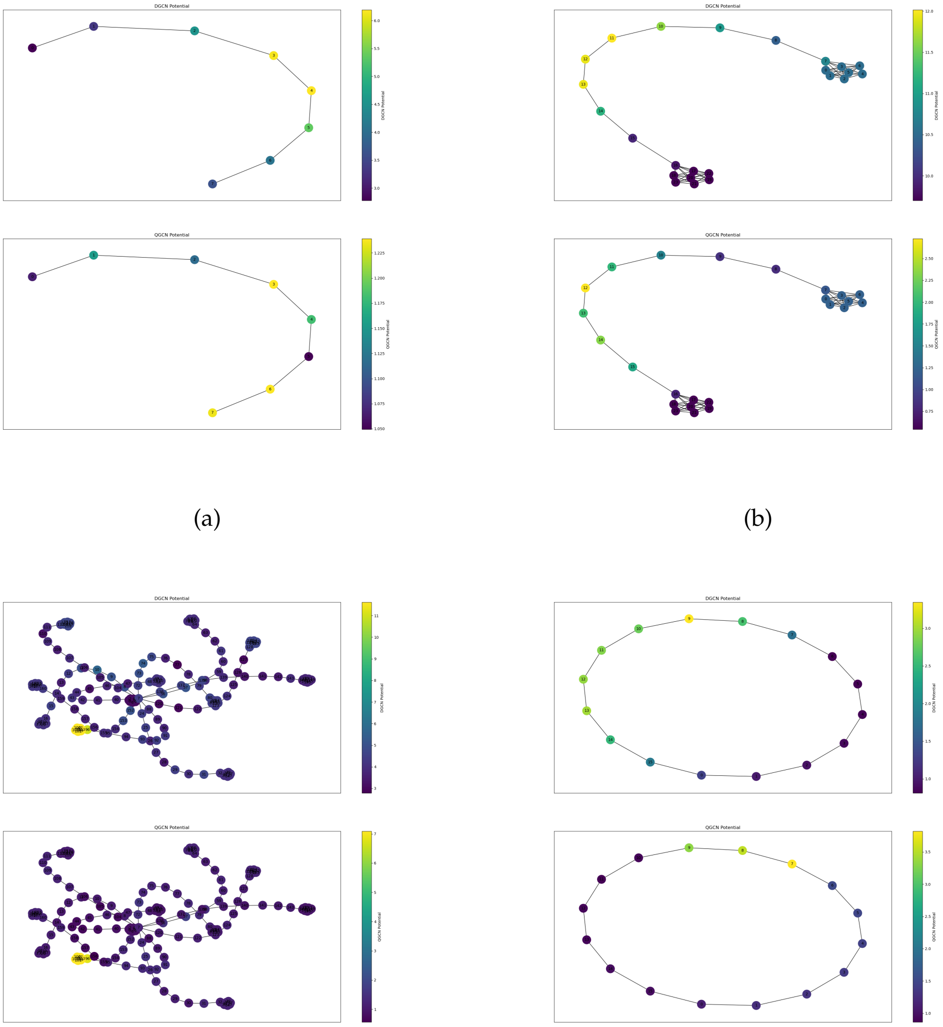
(a) (b)

(c) (d)

Figure 4: Test accuracy across synthetic tasks as a function of the number of layers. (a) Path Graph, (b) Barbell Graph, (c) Barbell-Star Graph, and (d) Ring Graph. In all cases, CTQW and CTQW+μ consistently outperform all other architectures.

Interpreting Learned Potentials

In biased models, we visualize the learned potentials across all tasks. In each case, 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 in the Ring task, μ successfully suppresses one of the two paths, concentrating signal flow along a single route. Beyond improving predictive performance, the learned potentials offer transparent, interpretable insights into how the model routes information across complex structures.



(a) (b)

(c) (d)

Figure 5: Visualization of the learned node-level bias μ on representative samples. (a) Path Graph, (b) Barbell Graph, (c) Barbell-Star Graph, and (d) Ring Graph. For each task, the top panel shows values of μ under the heat kernel, while the bottom panel shows those under CTQW. In all cases, μ highlights the most efficient route from the answer node to the query node, with the Ring task illustrating how one path is suppressed in favor of a single signal flow.