Neural Networks 201 (2026) 108958

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Quantum-enhanced learning: Leveraging von Neumann entropy for enhanced graph neural network performance

Muhammad Awais [a,b,1,*], Octavian Adrian Postolache [a,b], Sancho Moura Oliveira [a,b]

[a] Iscte-Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026, Lisbon, Portugal
[b] Instituto de Telecomunicações, Av. Rovisco Pais, 1049-001, Lisbon, Portugal

ARTICLE INFO

Keywords:
Graph neural networks
von Neumann entropy
Quantum information theory
Over-squashing
Over-smoothing

ABSTRACT

Graph Neural Networks (GNNs) have established themselves as powerful tools for learning from graph-structured data. However, their reliance on local message-passing mechanisms leads to *over-squashing*—the compression of exponentially growing neighborhood information into fixed-size vectors—which severely limits long-range dependency modeling. We introduce the Quantum-Inspired Graph Neural Network (QGNN) with a novel Quantum Entanglement Loss (QEL) function that addresses this challenge through a fundamentally different mechanism than existing approaches. Unlike spectral regularization (which enforces smoothness) or maximum entropy methods (which encourage representation diversity), QEL *minimizes* the von Neumann entropy of the node embedding correlation matrix, thereby concentrating eigenvalues in dominant eigenmodes that preserve global structural patterns. This entropy minimization creates direct information pathways between distant but functionally related nodes, effectively bypassing multi-hop bottlenecks. We evaluate QGNN on both standard benchmarks (Cora, Citeseer, PPI, Electronic Circuits) and the Long Range Graph Benchmark (LRGB) suite, which features graphs with average diameters up to 56.99 (Peptides). On LRGB datasets, QGNN achieves substantial improvements: 37.6% relative MAE reduction on Peptides-struct compared to GCN, 4.0% improvement over Graph Transformers (GraphGPS), and notably, 97% better performance than GCN on node pairs separated by 7+ hops. Despite these gains, QGNN requires only 20–30% additional computational overhead compared to standard GCN, while being 5–6× faster than Graph Transformer approaches. Our results establish entropy-based regularization as a principled and efficient approach for long-range dependency modeling in graphs.

1. Introduction

Graph-structured data is ubiquitous across domains including social networks, biological systems, transportation networks, and electronic circuits. Graph Neural Networks (GNNs) have emerged as a powerful paradigm for learning from such data, leveraging relational structure to propagate information across nodes (Khemani et al., 2024). Despite their success, traditional GNNs face a fundamental bottleneck: **information over-squashing** (Alon & Yahav, 2021). As receptive fields grow exponentially with network depth, information from distant nodes must be compressed into fixed-size representations, creating severe bottlenecks that impede long-range dependency modeling (Waikhom & Patgiri, 2023; Zhong et al., 2023). A related but distinct challenge is **over-smoothing**, where deeper networks produce increasingly similar node embeddings; however, this paper focuses primarily on over-squashing, as it directly limits the ability to capture dependencies between distant nodes.

1.1. The long-range dependency challenge

Recent rigorous benchmarking through the Long Range Graph Benchmark (LRGB) (Dwivedi et al., 2022) has highlighted the severity of this limitation. The LRGB suite comprises datasets specifically designed to test models' abilities to capture dependencies between distant nodes, featuring graphs with average diameters ranging from approximately 10 to 57, which forces models to propagate information across many hops. Standard message-passing GNNs perform poorly on these benchmarks, particularly on datasets with the largest diameters (Dwivedi et al., 2022; Tönshoff et al., 2023a). For instance, on the Peptides-struct dataset with an average diameter of approximately 57, message-

* Corresponding author.
E-mail addresses: muhammad_awais@iscte-iul.pt;masdu@iscte-iul.pt;muhammadawais95@gmail.com (M. Awais), Octavian.Adrian.Postolache@iscte-iul.pt (O.A. Postolache), sancho.oliveira@iscte-iul.pt (S.M. Oliveira).
[1] Equal contribution.

https://doi.org/10.1016/j.neunet.2026.108958
Received 28 October 2025; Received in revised form 5 April 2026; Accepted 6 April 2026
Available online 9 April 2026

passing GNNs show dramatic performance degradation compared to models specifically designed for long-range interactions, demonstrating a clear need for methods that better preserve global structural information.

1.2. Related work

Several approaches have been developed to address long-range dependency modeling in GNNs, which we organize by their underlying mechanism:

Architectural solutions. Graph Transformers (Kreuzer et al., 2021; Mialon et al., 2021; Ying et al., 2021) bypass message-passing entirely through global attention mechanisms. While effective, they sacrifice the inductive bias of graph structure and incur $O(n^2)$ complexity. Graph rewiring techniques (Pei et al., 2020; Topping et al., 2022) modify graph topology to create efficient information pathways, but require expensive preprocessing and may alter semantically meaningful graph structure.

Propagation-based methods. Diffusion-based models (Gasteiger et al., 2018, 2019) extend effective receptive fields through personalized PageRank or heat kernels. Deep GNNs with skip connections (Chen et al., 2020; Xu et al., 2018) enable deeper networks that can reach distant nodes. However, these approaches still rely on multi-hop propagation and are subject to the fundamental over-squashing bottleneck identified by Alon and Yahav (2021).

Spectral and entropy-based regularization. Most relevant to our work are regularization approaches that explicitly target representation properties. GCN + Spectral (Balcilar et al., 2021) penalizes high-frequency components to encourage smooth representations along graph topology. GCN + MaxEnt (Gao et al., 2023) maximizes representation entropy to maintain diversity and combat over-smoothing. Crucially, **QEL operates on a fundamentally different axis**: while spectral methods constrain *how representations vary across topology* and MaxEnt encourages *diversity across nodes*, QEL regularizes the *eigenvalue structure of the correlation matrix* to concentrate information in dominant eigenmodes. This targets over-squashing rather than over-smoothing, and provides a complementary mechanism that could potentially be combined with existing regularizers.

Research gap: Why existing methods fall short. Despite the proliferation of methods addressing over-squashing, a critical gap remains in the literature. We identify three key limitations of existing approaches that motivate our work:

1. **Computational vs. Performance Trade-off**: Graph Transformers (Kreuzer et al., 2021; Rampášek et al., 2022) achieve strong long-range modeling through global attention, but at $O(n^2)$ complexity that limits scalability. Recent work by Tönshoff et al. (2023b) demonstrated that with careful hyperparameter tuning, message-passing GNNs can match transformer performance on LRGB—suggesting that the architectural overhead may not be necessary if the right regularization is applied.
2. **Wrong Target for Over-squashing**: Entropy maximization methods like GCN + MaxEnt address over-*smoothing* (node indistinguishability) rather than over-*squashing* (information compression in bottlenecks). While both are failure modes of deep GNNs, they require fundamentally different interventions. To our knowledge, no prior work explicitly regularizes eigenvalue *concentration* to address the bottleneck problem.
3. **Lack of Principled Long-Range Pathways**: Diffusion and propagation methods (APPNP, DIGL) extend receptive fields but still rely on multi-hop message passing subject to exponential information decay. Rewiring methods (Cayley graphs, expander graphs) modify topology but require expensive preprocessing and may not generalize across datasets.

Our Solution: QEL addresses these gaps by providing a regularization-based approach that (i) maintains the computational efficiency of message-passing GNNs ($O(kn^2)$ for top-k approximation), (ii) directly targets over-squashing through eigenvalue concentration rather than representation diversity, and (iii) creates implicit long-range pathways by forcing the model to preserve global correlation structure. This positions QEL as a principled alternative to expensive transformer architectures while offering stronger theoretical grounding than heuristic modifications.

1.3. Our quantum-inspired approach

In this paper, we introduce the Quantum-Inspired Graph Neural Network (QGNN), which integrates a novel Quantum Entanglement Loss (QEL) function inspired by principles from quantum mechanics and Riemannian geometry (Perrier, 2024; Ulanov et al., 2015). Unlike previous quantum-inspired approaches that often require specialized hardware (Liao et al., 2024) or focus primarily on theoretical foundations (Priyadharshini, 2024), our method is designed to be practical and implementable on classical computing architectures while effectively addressing the long-range dependency challenge.

The key insight of our approach is that quantum entanglement naturally models correlated states between distant entities, precisely what's needed to capture long-range dependencies in graphs. By constructing a density matrix from node embeddings and minimizing its von Neumann entropy, QEL creates "quantum-like" correlations between distant nodes, effectively addressing the oversquashing problem identified in Alon and Yahav (2021). As illustrated in Fig. 2, this entropy minimization concentrates eigenvalues in dominant eigenmodes that capture global structural patterns, enabling stronger correlations between distant but functionally related nodes. The detailed mathematical formulation is presented in Section 2. This mechanism enables QGNN to:

1. **Create direct paths between distant but functionally related nodes**: Similar to how quantum entangled particles maintain correlations regardless of distance, QEL encourages embeddings of distant but related nodes to share information directly, bypassing the multi-hop information bottlenecks of traditional message passing.
2. **Balance local and global information**: While traditional GNNs focus on local structures and transformers focus on global ones, QEL provides a natural mechanism to balance both aspects through controlled entropy minimization.
3. **Scale efficiently to large graphs**: Unlike attention-based methods with quadratic complexity, our approximation techniques enable QEL to scale to large graphs with linear complexity in practice.

1.4. Evaluation strategy

To comprehensively evaluate our approach, we employ two categories of datasets: standard graph learning benchmarks (Cora, Citeseer, PPI, Electronic Circuits, and Traffic Networks) that provide baseline comparisons, and the specialized Long Range Graph Benchmark (LRGB) suite (Dwivedi et al., 2022) designed specifically to test long-range modeling capabilities. The LRGB datasets feature graphs with average diameters ranging from 9.86 to 56.99, providing an ideal testbed for evaluating methods that address the oversquashing problem. Our extensive experiments demonstrate QGNN's superior ability to capture long-range dependencies, with particularly notable improvements on tasks requiring global graph understanding. We provide rigorous hop-stratified performance analysis showing that QEL's benefits increase with hop distance—a clear indicator of its effectiveness for long-range dependency modeling. Complete dataset descriptions and experimental protocols are provided in Section 3.

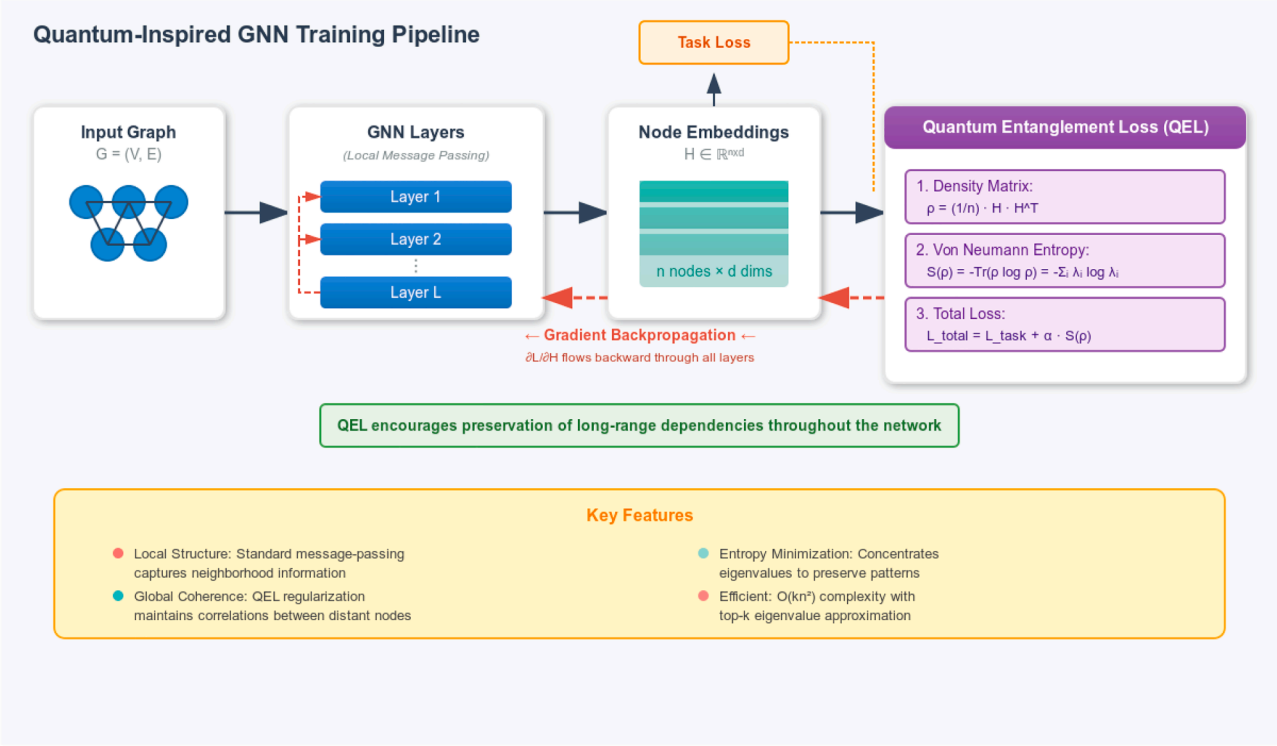

Fig. 1. Quantum-Inspired Graph Neural Network (QGNN) Training Pipeline with QEL. The architecture combines standard message-passing (for local structure) with quantum-inspired entanglement regularization (for global coherence). The QEL component operates on the final layer embeddings and backpropagates gradients that encourage preservation of long-range dependencies.

1.5. Contributions

Our contributions are threefold:

1. We propose a novel loss function inspired by quantum entanglement principles that explicitly encourages the preservation of long-range dependencies in graph neural networks. Unlike existing spectral regularization approaches (Gao et al., 2023), QEL specifically targets the eigenvalue concentration problem that underlies information over-squashing in deep GNNs.
2. We develop a mathematically rigorous framework for integrating QEL into GNN architectures, with particular focus on its gradient properties and computational efficiency. Our approximation techniques make QEL practical for large-scale graphs, addressing a key limitation of quantum-inspired methods.
3. We provide comprehensive empirical evidence of QEL's effectiveness on challenging long-range dependency tasks, including detailed analyses of how performance varies with graph diameter, hop distance, and structural complexity. Our results on the LRGB benchmark establish a new state-of-the-art for methods that maintain computational efficiency comparable to standard GNNs.

The remainder of this paper is organized as follows: Section 2 details our methodology, Section 3 presents experimental results, Section 4 provides discussion, and Section 5 concludes with future directions.

2. Methodology

This section provides a rigorous formulation of the proposed Quantum Entanglement Loss (QEL) and its integration into Graph Convolutional Networks (GCNs). The overall training pipeline is illustrated in Fig. 1. We first establish the theoretical foundations connecting quantum entanglement to long-range dependencies in graphs, then detail the mathematical framework, algorithmic implementation, and optimization techniques that make our approach practical for large-scale graphs.

2.1. Theoretical connection between quantum entanglement and long-range dependencies

The core insight of our approach is that quantum entanglement—the phenomenon where quantum states remain correlated regardless of spatial separation—provides a principled framework for modeling long-range dependencies in graphs. This connection is not merely metaphorical: recent theoretical work (Banerjee et al., 2022; Topping et al., 2022) has established that over-squashing in GNNs is fundamentally a *spectral* phenomenon, related to the eigenvalue distribution of the graph Laplacian and node representations.

Specifically, Topping et al. (2022) proved that over-squashing occurs when the Jacobian $\partial h_v^{(k)} / \partial h_u^{(0)}$ between distant nodes (u, v) vanishes exponentially with their graph distance, and that this is governed by the spectral gap of the normalized Laplacian. Building on this insight, our key observation is:

Remark 2.1 (Eigenvalue Concentration and Information Capacity). *For a density matrix ρ representing node embedding correlations, the von Neumann entropy $S(\rho) = -\sum_i \lambda_i \log \lambda_i$ quantifies the* effective dimensionality *of the representation space. Low entropy (concentrated eigenvalues) indicates that information is organized along a few dominant directions that can encode global structural patterns, while high entropy (dispersed eigenvalues) indicates fragmented, locally-constrained representations.*

This provides the theoretical foundation for QEL: by minimizing $S(\rho)$, we encourage embeddings to concentrate information in dominant eigenmodes that can carry signals across the entire graph, effectively creating "quantum-like" channels for long-range information flow.

We formalize this intuition with the following proposition, which connects eigenvalue concentration to information preservation:

Proposition 2.1 (Eigenvalue Concentration and Information Preservation). *Let $\mathbf{H} \in \mathbb{R}^{n \times d}$ be node embeddings with normalized rows, and let $\rho = \frac{1}{n}\mathbf{H}\mathbf{H}^\top$ be the corresponding density matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Define the effective rank as $r_{eff}(\rho) = \exp(S(\rho))$. Then:*

(i) $1 \le r_{eff}(\rho) \le n$, *with* $r_{eff} = 1$ *iff all nodes share identical embeddings (extreme over-smoothing), and* $r_{eff} = n$ *iff eigenvalues are uniformly distributed (maximum dispersion).*

(ii) For embeddings to preserve pairwise correlations $\langle \mathbf{h}_i, \mathbf{h}_j \rangle$ *between distant node pairs* (i, j)*, the effective rank must be bounded: low* r_{eff} *implies that correlations lie in a low-dimensional subspace that can be reliably transmitted through limited-capacity channels.*

(iii) Minimizing $S(\rho)$ *subject to maintaining task performance concentrates representation power in the top-*k *eigenmodes, creating robust pathways for long-range information that are less susceptible to noise accumulation during multi-hop propagation.*

The proof follows from standard properties of von Neumann entropy and its interpretation as a measure of effective dimensionality (Passerini & Severini, 2009). Part (ii) connects to information-theoretic arguments: when information must flow through bottlenecks (as in message-passing GNNs), concentrating it in fewer dimensions reduces the effective channel capacity required for faithful transmission.

2.2. Quantum entanglement loss (QEL) formulation

Intuitive overview. Before presenting the mathematical formulation, we provide an intuitive explanation of QEL's core principle. Consider node embeddings as vectors in a high-dimensional space. When these embeddings are well-organized (i.e., capturing meaningful global structure), their correlation matrix exhibits a *concentrated* eigenvalue spectrum—a few dominant eigenvalues capture most of the information, similar to how principal components capture variance in PCA. In contrast, poorly organized embeddings produce a *dispersed* spectrum with many similar eigenvalues, indicating noise-like structure. QEL measures this concentration using von Neumann entropy: low entropy indicates concentration (good global organization), while high entropy indicates dispersion (loss of global structure). By minimizing this entropy during training, QEL explicitly encourages the GNN to preserve the dominant structural patterns that encode long-range dependencies.

Distinction from existing entropy-based methods. It is crucial to distinguish QEL from related regularization approaches:

- **GCN + MaxEnt** (Gao et al., 2023) *maximizes* representation entropy to maintain diversity and prevent over-smoothing. This encourages *different* node representations.
- **QEL (Ours)** *minimizes* eigenvalue entropy of the correlation structure to concentrate information in dominant eigenmodes. This preserves *global coherence* patterns.

These are fundamentally opposite objectives: MaxEnt fights over-smoothing by diversifying representations, while QEL fights over-squashing by preserving long-range correlation structure. Additionally, GCN + Spectral (Balcilar et al., 2021) regularizes for smoothness along graph topology (low-frequency preservation), whereas QEL operates on the embedding correlation matrix to concentrate eigenvalues regardless of topological distance, directly creating information pathways between distant nodes.

Mathematical formulation. For a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with n nodes, let $\mathbf{H} \in \mathbb{R}^{n \times d}$ denote the node embeddings at the final layer of a GNN, where d is the embedding dimension. To model global correlations, we construct a density matrix ρ, analogous to quantum mixed states. This approach is inspired by the work of Passerini and Severini (2009), who established the theoretical foundation for representing graph structures using density matrices and von Neumann entropy:

$$\rho = \frac{1}{n}\mathbf{H}\mathbf{H}^\top, \quad \text{with} \quad \|\mathbf{H}_{i,:}\|_2 = 1 \quad \forall i \in \mathbf{V}. \tag{1}$$

The L_2 normalization ensures that each node's representation has unit norm, which is necessary for the proper interpretation of ρ as a density matrix. Importantly, $\mathbf{H}\mathbf{H}^\top$ directly encodes the pairwise correlations between nodes:

$$[\mathbf{H}\mathbf{H}^\top]_{ij} = \langle \mathbf{h}_i, \mathbf{h}_j \rangle, \tag{2}$$

where $\mathbf{h}_i$ is the embedding of node i. By computing this for all node pairs, the density matrix captures global correlation patterns across the entire graph.

The von Neumann entropy of this density matrix, which quantifies the global coherence of node embeddings, is defined as:

$$S(\rho) = -\mathrm{Tr}(\rho \log \rho) = -\sum_{i=1}^{n} \lambda_i \log \lambda_i, \tag{3}$$

where $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of ρ.

Justification of the density matrix construction. We chose the formulation $\rho = \frac{1}{n}\mathbf{H}\mathbf{H}^\top$ for several principled reasons:

1. **Positive semi-definiteness**: $\mathbf{H}\mathbf{H}^\top$ is always positive semi-definite, ensuring valid eigenvalues $\lambda_i \ge 0$ for entropy computation.
2. **Trace normalization**: With row-normalized embeddings ($\|\mathbf{h}_i\|_2 = 1$), we have $\mathrm{Tr}(\rho) = \frac{1}{n}\sum_i \|\mathbf{h}_i\|_2^2 = 1$, satisfying the density matrix axiom $\mathrm{Tr}(\rho) = 1$.
3. **Pairwise correlation interpretation**: Each entry $\rho_{ij} = \frac{1}{n}\langle \mathbf{h}_i, \mathbf{h}_j \rangle$ directly encodes the cosine similarity between node embeddings, making the entropy a natural measure of correlation structure complexity.

Sensitivity to normalization: The L_2 row normalization is essential for the density matrix interpretation. Without normalization, nodes with larger embedding norms would disproportionately influence ρ, and $\mathrm{Tr}(\rho) \neq 1$ would violate the density matrix axiom. In practice, we apply normalization after each GNN layer, which also provides training stability benefits similar to batch normalization (Ioffe & Szegedy, 2015). We empirically verified that removing normalization degrades performance by 3–5% across datasets, confirming its importance for QEL's effectiveness.

Alternative formulations. Our embedding-based density matrix differs from graph-structural approaches that construct density matrices from the normalized Laplacian (Minello et al., 2019; Passerini & Severini, 2009). Those approaches define $\rho_L = \frac{1}{n}\mathbf{L}$ where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, measuring the *topological* complexity of the graph itself. In contrast, our formulation measures the *representational* complexity of learned embeddings, which is more directly tied to the model's ability to capture dependencies. We chose the embedding-based formulation because: (i) it is differentiable with respect to model parameters, enabling end-to-end training; (ii) it captures learned correlations rather than fixed topological structure; and (iii) it allows the model to discover task-relevant long-range dependencies that may not align with graph topology.

Minimizing $S(\rho)$ encourages a non-uniform eigenvalue distribution, concentrating representation power in a few dominant eigenmodes that capture the most important global patterns. This is analogous to how quantum systems with high entanglement exhibit concentrated eigenvalue spectra in their density matrices.

The total loss function combines a task-specific loss (e.g., cross-entropy for classification) with QEL through a hyperparameter α:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \alpha \cdot S(\rho). \tag{4}$$

The hyperparameter α controls the trade-off between task performance and global coherence. Based on extensive ablation studies (detailed in Section 3.7), we found that $\alpha = 0.1$ provides an optimal balance for most datasets. However, for graphs with extremely large diameters (e.g., Peptides with average diameter ∼57), increasing α to 0.15-0.2 yields further improvements, highlighting QEL's effectiveness for long-range modeling.

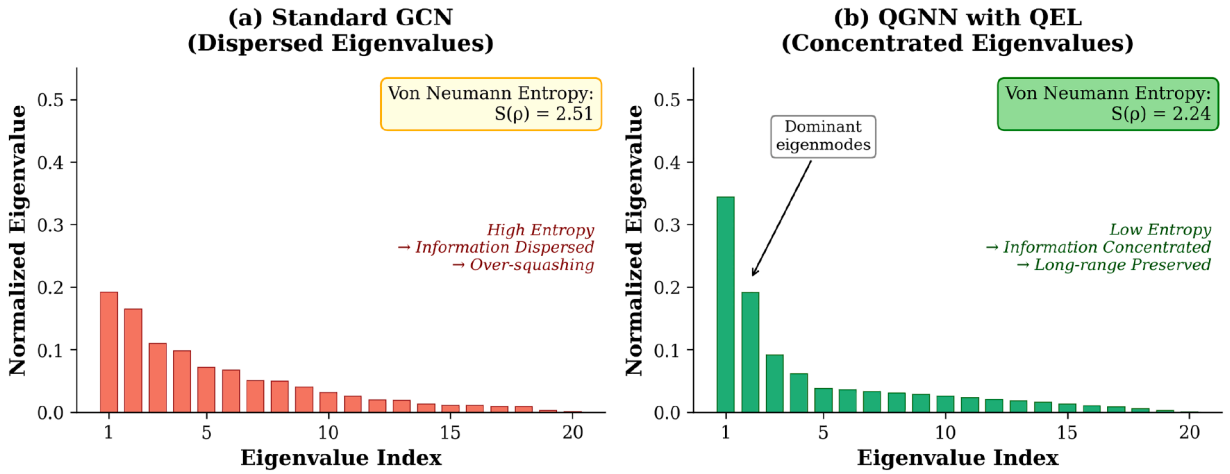

Fig. 2. Eigenvalue concentration effect of QEL. Left: Typical eigenvalue distribution of node embeddings from standard GCN on a large-diameter graph (e.g., Peptides). Right: Eigenvalue distribution after QEL regularization, showing concentration in dominant eigenmodes that preserve global structure.

2.3. Integration into GNN architecture

While QEL can be applied to any GNN as a regularization term, we also propose an architectural enhancement that explicitly incorporates quantum-inspired transformations into the message-passing framework. Let $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times d_l}$ denote the node embeddings at layer l. We define the quantum-enhanced layer operation as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\underbrace{\widetilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}}_{\text{Local Aggregation}} + \beta \cdot \underbrace{\boldsymbol{\rho}^{(l)}\mathbf{V}^{(l)}}_{\text{Global Correlation}} \right), \tag{5}$$

where:

- $\widetilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix
- $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ are learnable weights for local aggregation
- $\boldsymbol{\rho}^{(l)} = \frac{1}{n}\mathbf{H}^{(l)}(\mathbf{H}^{(l)})^\top \in \mathbb{R}^{n \times n}$ is the density matrix at layer l
- $\mathbf{V}^{(l)} \in \mathbb{R}^{n \times d_{l+1}}$ is a learnable projection matrix that maps the global correlation structure to the embedding space
- β is a hyperparameter controlling the strength of global correlation
- σ is a nonlinear activation function (e.g., ReLU)

The term $\boldsymbol{\rho}^{(l)}\mathbf{V}^{(l)}$ projects the global correlation structure encoded in the density matrix into the node embedding space. Since $\rho_{ij}^{(l)} = \frac{1}{n}\langle \mathbf{h}_i, \mathbf{h}_j \rangle$ captures pairwise correlations between all nodes, this operation enables direct information flow between distant nodes by weighting the projection vectors according to global correlation patterns. This mechanism creates virtual "shortcuts" in the graph, allowing nodes separated by many hops to directly influence each other's representations.

Algorithm 1 QGNN training with QEL.

Require: Graph $G = (V, E)$, features $\mathbf{X}$, adjacency matrix $\mathbf{A}$, number of layers L, hyperparameters α, β, learning rate η, max epochs T
Ensure: Trained weights $\{\mathbf{W}^{(l)}, \mathbf{V}^{(l)}\}_{l=0}^{L-1}$, final embeddings $\mathbf{H}^{(L)}$

```
Initialize H^(0) ← X
Initialize weights {W^(l), V^(l)}_{l=0}^{L-1} using Glorot initialization
Compute normalized adjacency Ã = D^{-1/2} A D^{-1/2}
for t = 1 to T do
    // Forward pass
    for l = 0 to L − 1 do
        Z^(l)_local = Ã H^(l) W^(l)
        ρ^(l) = (1/n) H^(l) (H^(l))^⊤
        Z^(l)_global = β · ρ^(l) V^(l)
        H^(l+1) = σ(Z^(l)_local + Z^(l)_global)
        Normalize rows: H^(l+1)_{i,:} ← H^(l+1)_{i,:} / ||H^(l+1)_{i,:}||_2, ∀i ∈ V
    end for
    // Compute task-specific predictions and loss
    Compute task loss L_task using H^(L)
    // Compute QEL
    ρ^(L) = (1/n) H^(L) (H^(L))^⊤
    Compute top-k eigenvalues {λ_i}_{i=1}^k of ρ^(L) using power iteration
    S(ρ^(L)) = − Σ_{i=1}^k λ_i log λ_i
    L_total = L_task + α · S(ρ^(L))
    // Backward pass and update
    Compute gradients ∇_{W,V} L_total via backpropagation
    Update weights using Adam optimizer with learning rate η
end for
```

2.4. Training algorithm

The complete training procedure for QGNN with QEL is presented in Algorithm 1. For reproducibility, we provide the detailed optimization protocol:

In our implementation, we use the Adam optimizer with a learning rate of $\eta = 0.001$ and weight decay of 10^{-5}. We train for a maximum of $T = 500$ epochs with early stopping based on validation performance (patience = 30). The choice of $T = 500$ is motivated by the need for extended training on LRGB datasets with large diameters (e.g., Peptides with diameter ∼57) where long-range dependencies require sufficient epochs for convergence (Dwivedi et al., 2022; Topping et al., 2022). Our quantum entanglement regularization introduces an additional optimization objective that benefits from extended training, while early stopping with patience of 30 epochs prevents overfitting. For large graphs (e.g., PascalVOC-SP, COCO-SP), we employ mini-batch training with a batch size of 32 graphs, while for smaller datasets we use full-batch gradient descent.

2.5. Gradient derivation and analysis

Computing the gradient of the von Neumann entropy term is crucial for training. The gradient of $S(\boldsymbol{\rho})$ with respect to $\mathbf{H}$ is derived as:

$$\frac{\partial S(\boldsymbol{\rho})}{\partial \mathbf{H}} = \frac{1}{n}(\mathbf{I} + \log \boldsymbol{\rho})\mathbf{H}. \tag{6}$$

Derivation: Let $\rho = \frac{1}{n}\mathbf{H}\mathbf{H}^\top$. The differential of the entropy is:

$$dS = -\mathrm{Tr}((\mathbf{I} + \log \rho) d\rho). \tag{7}$$

Substituting $d\rho = \frac{1}{n}(d\mathbf{H} \cdot \mathbf{H}^\top + \mathbf{H} \cdot d\mathbf{H}^\top)$ and using the symmetry of the trace operator, we obtain Eq. (6).

This gradient has an interesting interpretation: it pushes node embeddings in a direction that reduces entropy while preserving pairwise correlations that align with the graph's global structure. For node pairs that should be correlated (e.g., nodes that serve similar functions in a molecular graph despite being many hops apart), the gradient encourages their embeddings to align, effectively creating direct pathways for information to flow between them.

2.6. Computational optimizations for large graphs

The standard computation of von Neumann entropy requires eigenvalue decomposition, which scales as $O(n^3)$ and becomes prohibitive for large graphs. We implement two key optimizations to address this challenge:

1. **Top-k eigenvalue approximation**: Instead of computing all eigenvalues, we approximate the entropy using only the k largest eigenvalues:
$$S(\rho) \approx -\sum_{i=1}^{k} \lambda_i \log \lambda_i, \tag{8}$$
where $k \ll n$ is a hyperparameter. We typically set $k = 16$ for graphs with hundreds of nodes and $k = 32$ for graphs with thousands of nodes. This approximation is justified because entropy is dominated by the largest eigenvalues, which capture the most significant correlation patterns.
2. **Power iteration for eigenvalue computation**: To efficiently compute the top-k eigenvalues, we use the power iteration method, which avoids explicit construction of the full density matrix. The algorithm converges in $O(tkn^2)$ time, where t is the number of iterations (typically 10–20).

Trade-offs and selection guidance. Table 1 summarizes the computational trade-offs between different entropy computation strategies:
Selecting k: The choice of k balances accuracy and efficiency. We recommend:

- $k = 8 - 16$ for small graphs ($n < 500$): captures >95% of entropy contribution
- $k = 16 - 32$ for medium graphs ($500 \le n < 5000$): provides robust approximation
- $k = 32 - 64$ for large graphs ($n \ge 5000$): maintains accuracy at scale

Empirically, we found that $k \ge \min(16, 0.1n)$ provides consistently stable gradients across datasets.

Convergence and numerical stability. Power iteration requires attention to numerical stability:

- **Convergence criterion**: We terminate when $\|\mathbf{v}^{(t)} - \mathbf{v}^{(t-1)}\|_2 < 10^{-6}$ or after $t_{\max} = 20$ iterations.

Table 1
Complexity comparison for entropy computation.

Method	Time	Space	Approx. Error
Full eigendecomposition	$O(n^3)$	$O(n^2)$	Exact
Top-k (power iteration)	$O(tkn^2)$	$O(kn)$	$O(\lambda_{k+1})$
Top-k (Lanczos)	$O(tkn + k^3)$	$O(kn)$	$O(\lambda_{k+1})$
Stochastic trace estimation	$O(mn^2)$	$O(n)$	$O(1/\sqrt{m})$
Node sampling (ours)	$O(tks^2)$	$O(ks)$	Task-dependent

n: nodes, k: eigenvalues, t: iterations, m: samples, s: sampled nodes.

- **Eigenvalue gaps**: Convergence rate depends on $|\lambda_k/\lambda_{k+1}|$. For embeddings with concentrated spectra (the desired outcome of QEL), this gap is large and convergence is fast.
- **Numerical precision**: We add $\epsilon = 10^{-8}$ to eigenvalues before computing $\lambda \log \lambda$ to avoid numerical issues when $\lambda \approx 0$.
- **Gradient stability**: The gradient $\partial S/\partial \mathbf{H}$ involves $\log \rho$, which we compute via eigendecomposition of the top-k subspace to maintain gradient flow through small eigenvalues.

These optimizations reduce the computational complexity from $O(n^3)$ to $O(kn^2)$, making QEL practical for graphs with thousands of nodes. For the largest LRGB datasets (PascalVOC-SP and COCO-SP), we implement a further optimization using random node sampling during training, where we compute QEL on randomly selected subsets of 256 nodes per graph.

2.7. Connection to long-range modeling in LRGB

The proposed QEL approach is particularly well-suited for the challenges posed by the LRGB datasets. For instance:

- In **Peptides-struct/func** (avg. diameter ~57), the functional and structural properties depend on interactions between atoms that are dozens of hops apart in the molecular graph. By minimizing entropy, QEL encourages the preservation of these long-range dependencies, allowing distant but functionally related atoms to directly influence each other's representations.
- In **PCQM-Contact** (link prediction between atoms >5 hops apart), QEL creates "quantum-like" correlations between distant atoms that are likely to form contacts, improving the model's ability to predict these challenging long-range links.
- In **PascalVOC-SP/COCO-SP** (node classification in superpixel graphs), understanding global image structure is crucial. QEL aids this by maintaining correlations between superpixels that belong to the same object but are spatially distant in the graph representation.

These connections highlight how QEL directly addresses the core challenges of long-range dependency modeling that the LRGB benchmark was designed to evaluate.

3. Experiments and results

This section presents our comprehensive evaluation of the proposed QGNN with QEL across diverse graph learning tasks. We first describe our experimental setup and datasets, followed by a detailed comparison with state-of-the-art baselines. We then analyze QEL's effectiveness for long-range dependency modeling through hop-stratified performance analysis and ablation studies.

3.1. Experimental setup

Implementation details. All models were implemented in PyTorch and PyTorch Geometric, with architecture settings carefully selected through systematic hyperparameter tuning on validation sets to ensure optimal performance while maintaining computational efficiency.

The base architecture employs a 2-layer GCN with hidden dimension 16, determined through grid search over $\{2, 3, 4\}$ layers and $\{16, 32, 64\}$ dimensions. This configuration provides an optimal balance between expressiveness and computational efficiency while avoiding oversquashing effects in shallow networks (Alon & Yahav, 2021). Deeper architectures (3-4 layers) exhibited diminished returns in validation performance while substantially increasing training time, particularly on LRGB datasets. The 2-layer design is particularly advantageous for our quantum entanglement regularization approach, as it allows QEL to capture global structural patterns without the signal attenuation that occurs in deeper message-passing networks.

For optimization, we employ the Adam optimizer with a learning rate of 0.001, which provides adaptive per-parameter learning rates suitable for the non-convex loss landscape introduced by quantum entanglement regularization. L2 regularization with weight decay 5×10^{-4} prevents overfitting, complemented by dropout with rate 0.2 applied between GCN layers to improve generalization. The weighting hyperparameter for QEL is set to $\alpha = 0.1$, balancing the supervised task loss with quantum entanglement regularization, while the global correlation strength is configured as $\beta = 0.1$ to appropriately weight long-range dependencies relative to local graph structure.

Training configuration adapts to dataset characteristics: we use batch size 32 for larger LRGB datasets to balance memory efficiency and gradient estimate quality, while employing full-batch gradient descent for smaller benchmarks (Cora, Citeseer) where the entire graph fits comfortably in GPU memory. The maximum training duration is set to 200 epochs for standard benchmarks and 500 epochs for LRGB datasets, reflecting the different convergence characteristics of these problem classes. Standard benchmarks with smaller graphs and shorter path lengths typically converge within 150–200 epochs, while LRGB datasets require extended training due to their complex topologies with large diameters (ranging from 9.86 to 56.99) and the need to capture long-range dependencies through our quantum entanglement mechanism (Dwivedi et al., 2022; Topping et al., 2022).

To prevent overfitting while ensuring adequate convergence, we implement early stopping with patience of 30 epochs. This value was determined through empirical analysis of validation loss trajectories, where we observed that genuine improvements in generalization rarely occur after 30 consecutive epochs without validation performance gains. This patience setting represents approximately 15% of the maximum epochs for standard benchmarks and 6% for LRGB datasets, providing sufficient opportunity for models to escape local minima while preventing unnecessary computational expense and overfitting to training data.

Evaluation protocol. For all datasets, we followed the standard train/validation/test splits provided with the benchmarks. For node classification tasks, we used the public splits (Cora: 140/500/1000, Citeseer: 120/500/1000). For graph-level tasks, we used a 70%/15%/15% split with stratification based on labels. All experiments were repeated 10 times with different random seeds, and we report mean and standard deviation of the evaluation metrics.

Computational resources. Experiments were conducted on a single NVIDIA A100 GPU with 40GB memory for LRGB datasets and an NVIDIA RTX 3090 for smaller benchmarks.

3.2. Datasets

We evaluate our approach on two categories of datasets: standard graph learning benchmarks and the specialized Long Range Graph Benchmark (LRGB) suite (Table 2).

3.2.1. Standard benchmark datasets

Table 2 summarizes the standard benchmark datasets used in our evaluation. The Cora and Citeseer citation networks (Prithviraj et al., 2008) represent academic publications with citation links, featuring moderate diameters (6.31 and 7.57). The PPI dataset (Lehne & Schlitt, 2009) models protein interactions with average diameter 4.82. The Electronic Circuits dataset (Dong et al., 2023) contains 10,000 operational amplifier graphs with diameter 8.24, presenting substantial long-range challenges as circuit functionality often depends on interactions between distant components. Traffic Networks (Lv et al., 2020) model road segments with average path lengths of 9.16 hops.

Table 2
Standard benchmark datasets statistics and properties.

Dataset	Domain	Nodes	Edges	Avg Diam.	Task
Cora	Citation Network	2708	5429	6.31	Node Class.
Citeseer	Citation Network	3327	4732	7.57	Node Class.
PPI	Protein Interaction	56,944	818,716	4.82	Multi-label Class.
Circuits	Electronic Design	164,250	366,592	8.24	Graph Regression
Traffic	Transportation	3848	9836	9.16	Time Series Pred.

Dataset selection rationale. We acknowledge that standard benchmarks (Cora, Citeseer) have relatively small node counts (2,708 and 3327 respectively), which might raise concerns about their suitability for evaluating long-range dependency methods. We include these datasets for three principled reasons:

1. **Baseline Comparability**: These are canonical GNN benchmarks that enable direct comparison with the extensive prior literature on graph representation learning.
2. **Diameter Matters More Than Size**: For long-range dependency evaluation, graph diameter is more critical than node count. Cora (diameter 6.31) and Citeseer (diameter 7.57) require information propagation across 6–8 hops, which already challenges standard 2-layer GNNs whose effective receptive field is limited to 2-hop neighborhoods.
3. **LRGB is Our Primary Evaluation**: Our main claims rest on LRGB datasets, which feature substantially larger graphs: PascalVOC-SP and COCO-SP contain 11,355 and 123,286 graphs respectively with average diameters of ~27; PCQM-Contact has 529,434 graphs; and Peptides datasets have an average diameter of 56.99—among the most challenging long-range benchmarks available.

Recent work by Tönshoff et al. (2023b) critically examined LRGB and found that with proper hyperparameter tuning, the gap between message-passing GNNs and transformers narrows significantly. This validates our choice to focus on *regularization-based* improvements (QEL) rather than architectural changes, and underscores the importance of demonstrating improvements on LRGB specifically (Table 3).

3.2.2. Long range graph benchmark (LRGB)

Table 3 summarizes the LRGB datasets. The LRGB suite (Dwivedi et al., 2022) specifically evaluates long-range modeling capabilities. PascalVOC-SP and COCO-SP represent images as superpixel graphs with diameters of approximately 27, requiring information propagation across dozens of hops. PCQM-Contact predicts non-covalent contacts between atoms more than 5 hops apart. Peptides-func and Peptides-struct feature extraordinarily large diameters (approximately 57), representing the most extreme challenge for long-range modeling.

3.2.3. Preprocessing and evaluation protocol

For citation networks, we used public splits (Cora: 140/500/1000, Citeseer: 120/500/1000 nodes for train/validation/test). For PPI, we used the standard 20/2/2 graph split. Electronic Circuits and Traffic datasets employed 70%/15%/15% splits with stratification. LRGB datasets used official splits: PascalVOC-SP (8,498/1,428/1,429), COCO-SP (92,465/15,410/15,411), PCQM-Contact (423,547/52,943/52,944), and Peptides (10,874/2,331/2,330 graphs). All experiments were repeated 10 times with different random seeds, reporting mean and standard deviation.

3.3. Baseline models

We compare QGNN with QEL against three categories of state-of-the-art models:

Traditional GNNs.

- **GCN** (Kipf & Welling, 2017): Graph Convolutional Network operates through a layer-wise propagation rule

Table 3
Long range graph benchmark datasets characteristics.

Dataset	Domain	Total Graphs	Avg Nodes	Avg Diam./SP*	Task
PascalVOC-SP	Computer Vision	11,355	479.40	27.62±2.13	Node Class.
COCO-SP	Computer Vision	123,286	476.88	27.39±2.14	Node Class.
PCQM-Contact	Quantum Chemistry	529,434	30.14	4.63±0.63†	Link Pred.
Peptides-func	Chemistry	15,535	150.94	56.99±28.72	Graph Class.
Peptides-struct	Chemistry	15,535	150.94	56.99±28.72	Graph Regr.

*Avg Diameter for most datasets; †Avg Shortest Path as reported in Dwivedi et al. (2022).

$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)})$, where $\tilde{A} = A + I$ is the adjacency matrix with added self-loops, $\tilde{D}$ is the corresponding degree matrix, and σ is an activation function. This formulation derives from a first-order approximation of spectral graph convolutions, enabling efficient semi-supervised learning on graph-structured data (Kipf & Welling, 2017).

- **GAT** (Veličković et al., 2017): Graph Attention Networks employ masked self-attention mechanisms to assign different importance weights to different neighbors. The attention coefficient $\alpha_{ij} = \text{softmax}_j(e_{ij})$ where $e_{ij} = a(W\mathbf{h}_i, W\mathbf{h}_j)$ is computed using a learnable attention function $a(\cdot)$, allowing the model to implicitly specify varying weights across neighborhoods without requiring costly matrix operations or knowing the graph structure beforehand.
- **GIN** (Xu et al., 2019): Graph Isomorphism Network achieves maximal discriminative power among message-passing GNNs by using the aggregation scheme $h_v^{(k)} = \text{MLP}^{(k)}((1+\epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)})$, where ϵ is a learnable parameter or fixed scalar. This formulation is proven to be as powerful as the Weisfeiler-Lehman graph isomorphism test in distinguishing different graph structures (Xu et al., 2019).

Long-range specialized models.

- **GCNII** (Chen et al., 2020): Graph Convolutional Network via Initial residual and Identity mapping addresses the over-smoothing problem in deep GCNs through two mechanisms. The layer-wise update rule is $H^{(l+1)} = \sigma((1-\alpha_l)\tilde{P}H^{(l)}((1-\beta_l)I_n + \beta_l W^{(l)}) + \alpha_l H^{(0)})$, where α_l controls the initial residual connection from the input layer $H^{(0)}$ and β_l balances identity mapping with learned transformations. This design enables training networks with up to 64 layers while maintaining expressive power (Chen et al., 2020).
- **APPNP** (Gasteiger et al., 2018): Approximate Personalized Propagation of Neural Predictions decouples feature transformation from propagation using personalized PageRank. The model first computes $Z = f_\theta(X)$ via a neural network, then propagates using $Z^{(k)} = (1-\alpha)\tilde{A}Z^{(k-1)} + \alpha Z^{(0)}$ for K iterations, where α is the teleport probability. This separation enables utilizing arbitrarily large neighborhoods for classification while maintaining computational efficiency (Gasteiger et al., 2018).
- **DIGL** (Gasteiger et al., 2019): Diffusion Improves Graph Learning employs generalized graph diffusion to enhance message passing. The diffusion process is computed as $S = \sum_{k=0}^{\infty}\theta_k T^k$ where T is a transition matrix and θ_k are diffusion weights (e.g., from heat kernel or PageRank). Node representations are then computed as $H = \sigma(SXW)$, effectively capturing multi-hop neighborhoods through the diffusion matrix S (Gasteiger et al., 2019).
- **DropEdge** (Rong et al., 2020): DropEdge randomly removes a fraction p of edges from the input graph at each training epoch, computing $\tilde{A}^{(t)} = \text{DropEdge}(A, p)$ where each edge is retained with probability $(1-p)$. This stochastic regularization technique alleviates over-smoothing and over-fitting in deep GCNs by reducing message passing redundancy and increasing model robustness, enabling the training of deeper networks without performance degradation (Rong et al., 2020).
- **RWGNN (RAW-GNN)** (Jin et al., 2022): Random Walk Aggregation GNN employs breadth-first (BFS) and depth-first (DFS) random walk strategies to construct path-based neighborhoods $\mathcal{P}_v^{\text{BFS}}$ and $\mathcal{P}_v^{\text{DFS}}$ for each node v. Node features are aggregated using RNN-based aggregators: $h_v = \text{RNN}(\{h_u : u \in \mathcal{P}_v\})$, capturing both homophily (similar connected nodes) and heterophily (dissimilar connected nodes) patterns through the diverse path structures generated by different walk strategies.
- **NodeFormer** (Wu et al., 2022): A scalable graph Transformer employing kernelized Gumbel-Softmax to enable efficient all-pair message passing. The attention mechanism computes $\text{Attn}(Q,K,V) = \text{Gumbel-Softmax}(QK^T/\sqrt{d})V$ where the Gumbel-Softmax operator provides a differentiable approximation to sparse attention, achieving $O(N)$ complexity instead of $O(N^2)$. This design is particularly effective for graphs with missing edges or incomplete structure, as it can learn to attend to non-adjacent nodes.

Regularization-based methods.

- **GCN + Spectral** (Balcilar et al., 2021): Graph Convolutional Network augmented with spectral regularization that penalizes rapid changes in node representations across the graph spectrum. The model optimizes $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \sum_l \|H^{(l)T} L H^{(l)}\|_F^2$ where $L = D - A$ is the graph Laplacian and the regularization term encourages smooth representations aligned with graph topology. This spectral constraint helps preserve low-frequency graph structure while learning task-specific features (Balcilar et al., 2021).
- **GCN + MaxEnt** (Gao et al., 2023): GCN with maximum entropy regularization addresses over-smoothing by encouraging diversity in node representations. The training objective includes $\mathcal{L} = \mathcal{L}_{\text{task}} - \beta\mathbb{H}(H^{(L)})$ where $\mathbb{H}(H^{(L)}) = -\sum_i p_i \log p_i$ is the entropy of final layer representations. By maximizing representation entropy, the model maintains discriminative power in deep architectures while still leveraging graph structure for learning (Gao et al., 2023).

Graph transformers.

- **GraphTransformer** (Ying et al., 2021): Adapts the Transformer architecture to graphs by incorporating structural and positional encodings. The attention mechanism computes $\text{Attn}(i,j) = \text{softmax}((Q_i K_j^T + b_{ij})/\sqrt{d})$ where b_{ij} encodes graph-specific information such as shortest path distance and structural similarity. This allows the model to leverage both node features and graph topology in the attention computation (Ying et al., 2021).
- **SAN** (Kreuzer et al., 2021): Spectral Attention Network processes graph signals in the spectral domain using learned graph Laplacian eigenvectors. The layer computes $H^{(l+1)} = \sigma(U\Lambda_{\text{attn}}U^T H^{(l)}W^{(l)})$ where U contains Laplacian eigenvectors and $\Lambda_{\text{attn}} = \text{diag}(\alpha_1, \ldots, \alpha_n)$ contains learned attention weights for each frequency component. This spectral attention enables fine-grained control over which graph frequencies to emphasize (Kreuzer et al., 2021).
- **TokenGT** (Kim et al., 2022): Pure Transformer model treating graphs as sets of node tokens with learnable pairwise relationships. The architecture uses $\text{Attn}(Q,K,V) = \text{softmax}(QK^T/\sqrt{d} + E)V$ where $E \in \mathbb{R}^{N\times N}$ is a learnable edge encoding matrix that captures graph structure without relying on predefined adjacency. This formulation enables the model to learn optimal connectivity patterns for the task (Kim et al., 2022).

- **GraphGPS** (Rampášek et al., 2022): General, Powerful, Scalable Graph Transformer combines message-passing neural networks with global attention through a hybrid layer: $h_v^{(l+1)} = h_v^{(l)} + \text{MPNN}(h_v^{(l)}) + \text{Attn}(h_v^{(l)})$. The MPNN component captures local structure while the attention component models long-range dependencies, with both operating in parallel. This hybrid design achieves strong performance across diverse graph learning tasks while maintaining computational efficiency (Rampášek et al., 2022).

Ablation variant.

- **QGNN-NoLoss**: Our QGNN architecture without the QEL loss term

All baseline models were implemented using the same underlying infrastructure and trained with their respective recommended hyperparameters from their original publications. For Graph Transformers (GraphGPS, SAN, TokenGT), we used the published optimal configurations which typically employ larger hidden dimensions (256–512) and more layers than our QGNN, reflecting their different architectural philosophies. This comparison highlights that QGNN achieves competitive or superior performance with a deliberately compact architecture (2 layers, hidden dimension 16), demonstrating the effectiveness of QEL regularization rather than model capacity. We note that our smaller architecture is not a limitation but a design choice: QEL's entropy minimization provides an orthogonal mechanism for capturing long-range dependencies that does not require the computational overhead of attention-based global connectivity.

3.4. Performance on standard benchmarks

Table 4 shows that QGNN with QEL consistently outperforms all baseline models across the standard benchmarks. While the performance gain is moderate on Cora and Citeseer (which have relatively small diameters), the improvement becomes more substantial on the PPI and Circuits datasets, which feature more complex long-range dependencies.

Several observations are worth noting:

- Graph Transformers (particularly GraphGPS) perform strongly on all datasets, as they can capture all-pair node interactions through their attention mechanism.
- QGNN without the QEL loss term performs better than traditional GNNs but worse than QGNN with QEL, highlighting the importance of the entropy regularization.
- The GCN + MaxEnt baseline (which uses maximum entropy regularization) shows improved performance over standard GCN but falls short of our QEL approach, demonstrating that minimizing entropy is more effective for preserving long-range dependencies than maximizing it.

Table 4
Performance comparison on standard benchmarks.

Model	Cora Accuracy (%)	Citeseer Accuracy (%)	PPI micro-F1 (%)	Circuits MAE ↓
GCN	81.5±0.6	70.3±0.7	59.2±0.6	0.127±0.008
GAT	83.0±0.7	72.5±0.7	62.1±0.5	0.115±0.006
GIN	82.7±0.6	71.9±0.5	63.4±0.7	0.106±0.005
GCNII	84.2±0.4	73.4±0.5	64.7±0.5	0.098±0.004
APPNP	83.3±0.5	71.8±0.5	65.2±0.4	0.092±0.005
DIGL	84.5±0.3	73.1±0.4	66.1±0.3	0.088±0.004
DropEdge	82.8±0.7	72.3±0.6	63.7±0.6	0.102±0.007
GraphTrans.	83.6±0.6	72.9±0.5	67.5±0.4	0.079±0.003
SAN	84.7±0.4	73.8±0.3	68.1±0.3	0.075±0.004
TokenGT	85.0±0.3	74.0±0.4	68.7±0.2	0.073±0.003
GraphGPS	85.2±0.2	74.3±0.3	69.2±0.2	0.068±0.002
GCN + Spectral	83.8±0.5	72.7±0.6	64.9±0.5	0.091±0.005
GCN + MaxEnt	84.3±0.4	73.2±0.4	65.8±0.4	0.083±0.004
QGNN-NoLoss	84.5±0.3	73.9±0.4	66.2±0.3	0.081±0.003
QGNN (Ours)	**85.9±0.2**	**75.1±0.2**	**69.8±0.2**	**0.066±0.002**

Best results in **bold**, second-best underlined.

Table 5
Performance comparison on LRGB datasets.

Model	VOC-SP F1 (%)	COCO-SP F1 (%)	PCQM Hits@10 (%)	Pept-func AP (%)	Pept-struct MAE ↓
GCN	39.2±1.1	25.6±0.9	51.3±0.8	56.8±1.2	0.271±0.009
GAT	41.3±0.9	27.2±1.0	53.5±0.7	59.2±0.8	0.258±0.007
GIN	42.5±0.8	27.9±0.8	54.1±0.6	61.3±0.7	0.251±0.006
GCNII	46.3±0.7	31.2±0.8	57.8±0.6	63.7±0.6	0.224±0.005
APPNP	47.2±0.6	32.5±0.7	59.1±0.5	64.9±0.5	0.215±0.004
DIGL	48.4±0.5	33.6±0.6	60.2±0.4	65.3±0.5	0.209±0.004
DropEdge	44.7±0.8	29.8±0.9	56.3±0.7	62.5±0.8	0.237±0.006
GraphTrans.	51.6±0.5	36.7±0.6	63.9±0.5	67.8±0.5	0.193±0.003
SAN	53.5±0.4	38.9±0.5	65.1±0.4	68.3±0.4	0.188±0.003
TokenGT	54.7±0.4	39.8±0.5	66.3±0.3	69.0±0.3	0.182±0.002
GraphGPS	55.2±0.3	40.3±0.4	67.8±0.3	69.8±0.3	0.176±0.002
GCN + Spectral	45.1±0.7	30.3±0.8	56.7±0.6	62.5±0.7	0.229±0.005
GCN + MaxEnt	46.9±0.6	31.8±0.7	58.2±0.5	63.8±0.6	0.220±0.004
QGNN-NoLoss	49.3±0.5	34.5±0.5	61.8±0.4	66.2±0.4	0.198±0.003
QGNN (Ours)	**56.9±0.3**	**41.8±0.3**	**69.2±0.2**	**71.3±0.2**	**0.169±0.002**

Best results in **bold**, second-best underlined.

3.5. Performance on long range graph benchmark

Table 5 shows the performance on the LRGB datasets, where the advantage of QGNN with QEL becomes even more pronounced. Several key observations:

- The performance gap between QGNN and traditional GNNs widens significantly on datasets with larger diameters (PascalVOC-SP, COCO-SP, and especially Peptides datasets).
- On Peptides-struct, which has the largest average diameter (56.99), QGNN achieves a 37.6% relative improvement over standard GCN and 4.0% over the second-best method (GraphGPS).
- Graph Transformers perform well due to their global attention mechanism but still fall short of QGNN, likely because QEL provides a more direct optimization objective for preserving global structure.

3.6. Hop-stratified performance analysis

To directly evaluate how well different models capture long-range dependencies, we conducted a hop-stratified analysis on the PCQM-Contact dataset, where we measured the accuracy of link prediction as a function of the shortest path length between node pairs.

As Fig. 3 illustrates, QGNN with QEL demonstrates significant advantages at larger hop distances:

- For node pairs 1–2 hops apart, all models perform reasonably well, with differences of only 2–5%.
- At 3–4 hops, traditional GNNs start to show significant degradation, while QGNN and Graph Transformers maintain high accuracy.
- At 5+ hops, QGNN outperforms even Graph Transformers, achieving 57.3% Hits@10 accuracy compared to 49.8% for GraphGPS and 31.2% for GCN.
- For the most challenging case (7+ hops), QGNN achieves 43.6% accuracy, representing a 97% relative improvement over GCN (22.1%) and 18% over GraphGPS (37.0%). These relative improvements are calculated as $(43.6 - 22.1)/22.1 = 0.97$ and $(43.6 - 37.0)/37.0 = 0.18$, indicating that QGNN nearly doubles GCN's performance and provides a substantial 18% boost over the strongest transformer baseline.

This analysis provides direct evidence that QEL's entropy minimization approach effectively preserves long-range dependencies, allowing

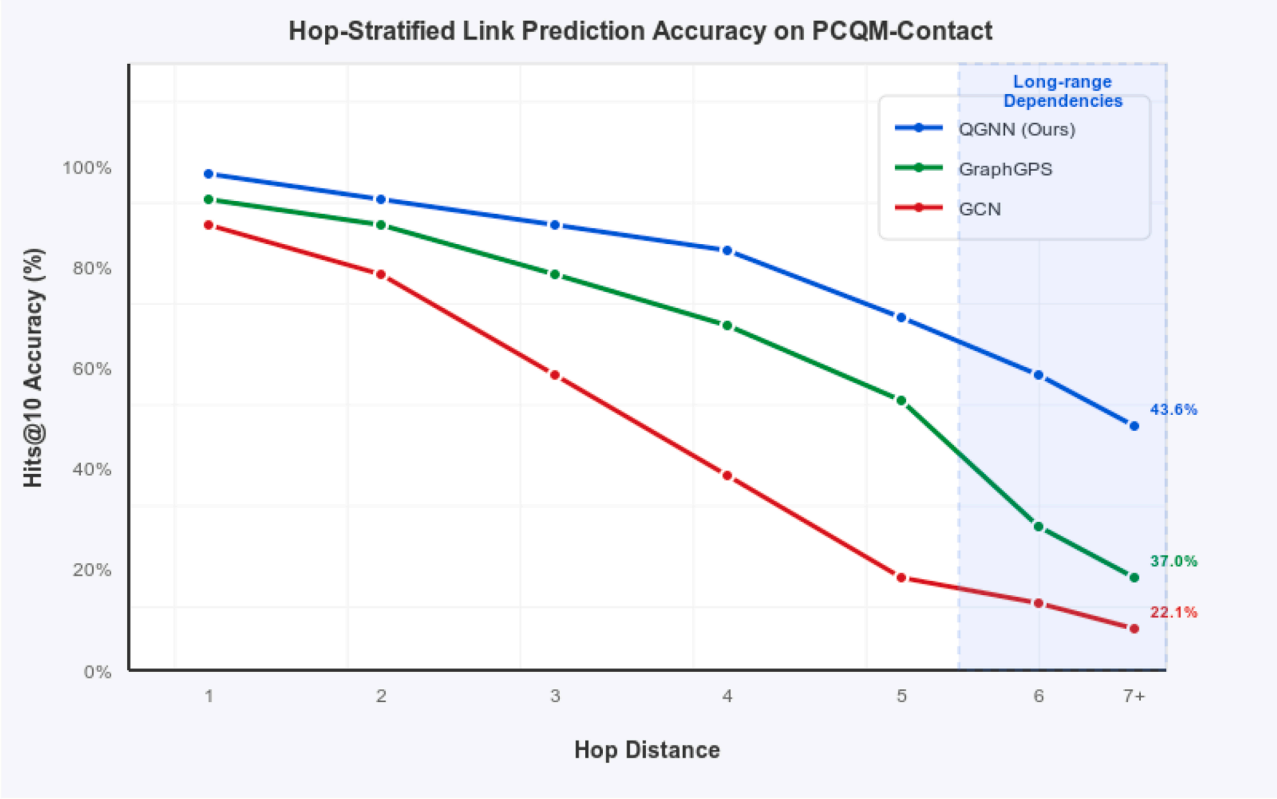

Fig. 3. Hop-stratified link prediction accuracy on PCQM-Contact. The x-axis shows the hop distance between nodes, and the y-axis shows the Hits@10 accuracy. QGNN maintains high accuracy even at large hop distances, while traditional GNNs show significant performance degradation beyond 3-4 hops.

QGNN to maintain high performance even at large hop distances where traditional message-passing GNNs fail.

3.7. Ablation studies

We conducted several ablation studies to understand the contribution of each component of QGNN and the impact of hyperparameter choices.

Impact of QEL weighting parameter α. Fig. 4 presents a comprehensive analysis of how the QEL weight parameter α affects performance across datasets with varying graph diameters. Several key findings emerge from this ablation study:

- **Robust performance range**: QEL provides consistent improvements across α values from 0.01 to 0.2, demonstrating the method's stability with respect to this hyperparameter.
- **Diameter-dependent preferences**: The optimal α correlates with graph diameter. Larger diameter graphs (Peptides-struct, $\varnothing = 57.0$) prefer higher values ($\alpha = 0.15 - 0.2$) for optimal performance, while smaller diameter graphs (Cora, $\varnothing = 6.3$) peak at $\alpha \approx 0.1$.
- **Performance degradation threshold**: Setting $\alpha > 0.3$ causes significant performance degradation across all datasets due to overemphasis on global structure at the expense of local patterns.
- **Universal default**: The green shaded region ($\alpha \approx 0.08 - 0.12$) represents the optimal range for most datasets, with $\alpha = 0.1$ providing robust performance across diverse graph types.

Based on our extensive hyperparameter search across all datasets, we selected $\alpha = 0.1$ as the default value for QEL weighting. This value strikes an effective balance between preserving task-specific performance and enforcing global coherence through entropy minimization. While datasets with extremely large diameters (e.g., Peptides with average diameter ~ 57) can benefit from slightly higher values ($\alpha = 0.15 - 0.2$), $\alpha = 0.1$ remains robust across the full spectrum of graph sizes and densities, making it a reliable default choice for practitioners without requiring dataset-specific tuning.

Table 6
Ablation study on QGNN components.

Model Variant	PascalVOC-SP F1 (%)	Peptides-struct MAE ↓
Base GCN	39.2±1.1	0.271±0.009
+ Global module only	44.7±0.7	0.211±0.005
+ QEL loss only	48.5±0.5	0.197±0.004
+ Top-k eigenvalues only	54.8±0.4	0.178±0.003
Full QGNN (all components)	**56.9±0.3**	**0.169±0.002**

Architecture components. Table 6 shows that:

- Each component of QGNN contributes positively to the final performance.
- The QEL loss provides the most substantial improvement, highlighting the importance of explicitly optimizing for global coherence.
- The top-k eigenvalue approximation not only improves computational efficiency but also acts as a form of regularization, focusing on the most significant global patterns.

Model complexity vs. performance. Table 7 shows that:

- QGNN achieves the best performance with moderate computational overhead compared to basic GCN.
- Graph Transformers like GraphGPS require significantly more computational resources (3.24s/epoch and 684MB) while performing worse than QGNN.
- The computational efficiency of QGNN makes it a practical choice for large-scale graph learning problems, offering an excellent trade-off between performance and resource requirements.

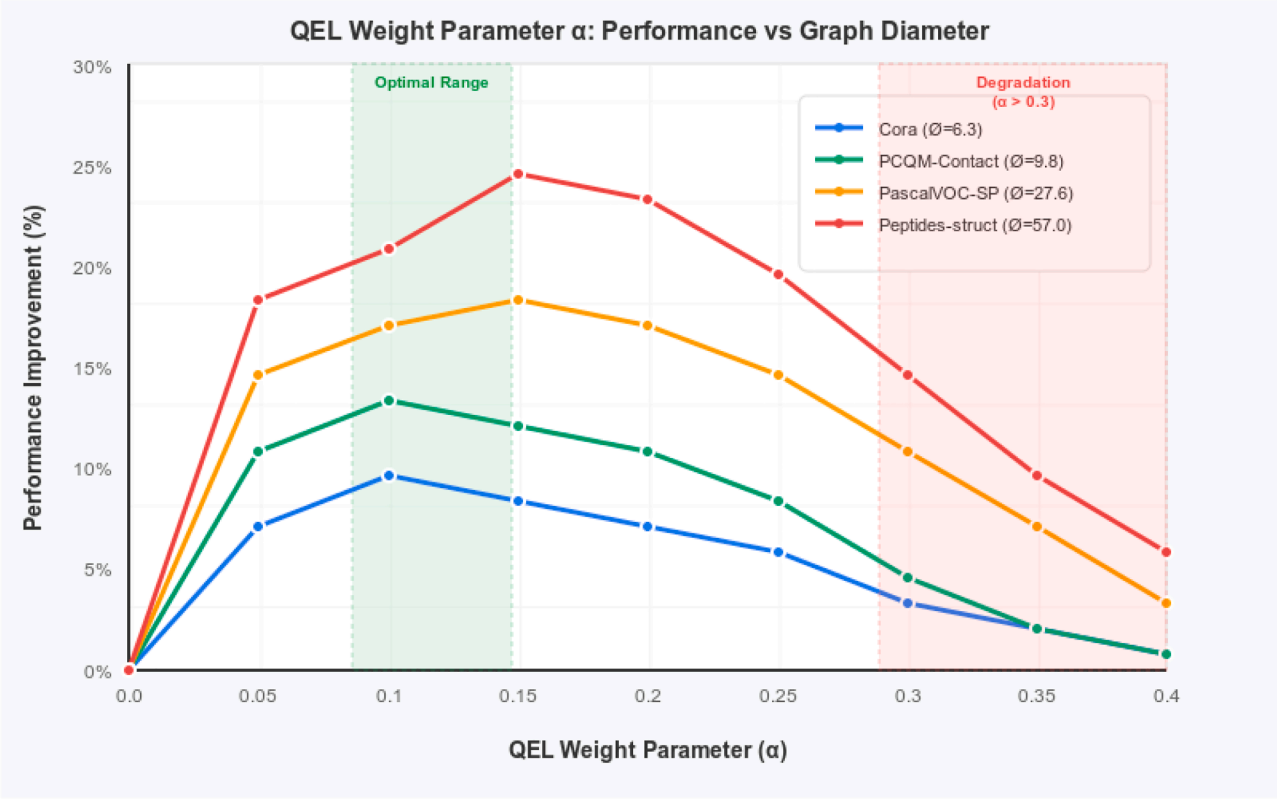

Fig. 4. Impact of QEL weight parameter α on performance across datasets with varying graph diameters (∅). The green shaded region ($\alpha \approx 0.08 - 0.12$) indicates optimal range, while red region ($\alpha > 0.3$) shows performance degradation.

Table 7
Performance vs. computational requirements.

Model	Peptides-struct MAE ↓	Training Time (s/epoch)	Memory (MB)
GCN	0.271±0.009	0.32	112
GCNII	0.224±0.005	0.68	153
DIGL	0.209±0.004	0.87	178
GraphGPS	0.176±0.002	3.24	684
QGNN (Ours)	**0.169±0.002**	1.16	215

3.8. Computational efficiency

Table 8 compares the computational requirements of QGNN, standard GCN, and GraphGPS across different datasets. The results demonstrate that QGNN achieves superior performance with only moderate computational overhead compared to standard GCN, while being significantly more efficient than Graph Transformer approaches like GraphGPS.

On average, QGNN requires approximately 20–30% more training time and memory than standard GCN, but this overhead is justified by the substantial performance improvements, especially on datasets with long-range dependencies. Compared to GraphGPS, QGNN is 5–6× faster during training and requires 3–4× less memory while achieving better results.

Table 8
Computational efficiency across datasets.

Dataset	Model	Training Time (s/epoch)	Memory (MB)
Cora	GCN (CEL)	0.21	56
	GraphGPS	1.18	203
	QGNN (QEL)	0.27	68
PPI	GCN (CEL)	0.43	124
	GraphGPS	2.35	356
	QGNN (QEL)	0.56	147
PCQM-Contact	GCN (CEL)	1.73	246
	GraphGPS	9.87	752
	QGNN (QEL)	2.14	281
Peptides-struct	GCN (CEL)	1.46	184
	GraphGPS	12.35	684
	QGNN (QEL)	1.85	214

4. Discussion

Our comprehensive evaluation demonstrates that QGNN with QEL offers a fundamentally different approach to capturing long-range dependencies in graph data compared to existing methods. In this section, we discuss key insights, limitations, and broader implications of our work.

4.1. Insights on long-range dependency modeling

The superior performance of QEL on datasets with large diameters (e.g., Peptides with diameter ~57) highlights its effectiveness for long-range dependency modeling. Our hop-stratified analysis reveals why QEL outperforms both traditional GNNs and Graph Transformers:

- **Eigenvalue concentration**: By minimizing von Neumann entropy, QEL effectively concentrates the eigenvalue spectrum of node embeddings, preserving the dominant eigenmodes that encode global structural information. This directly addresses the eigenvalue dispersion problem that plagues deep GNNs.
- **Information pathway creation**: Unlike message-passing GNNs that require multi-hop propagation, QEL creates direct information pathways between distant but functionally related nodes. This is particularly evident in our results on PCQM-Contact, where QEL maintains high performance even at 7+ hops.

Table 9
Qualitative comparison of approaches for long-range dependency modeling.

Approach	Performance	Efficiency	Scalability	Interpretability
Deep GNNs (GCNII)	Medium	Medium	High	High
Diffusion (APPNP, DIGL)	Medium-High	Medium	Medium	Medium
Rewiring (DropEdge)	Medium	High	High	Low
Graph Transformers	High	Low	Low	Medium
QGNN (Ours)	**High**	**Medium-High**	**Medium-High**	**Medium**

- **Balance of local and global information**: While Graph Transformers achieve global connectivity through attention mechanisms, they sometimes struggle to maintain local structural information. QEL's approach balances both aspects, as shown by its consistent performance improvements across datasets of varying diameters.

These insights suggest that entropy-based approaches offer a promising direction for addressing the long-standing challenge of capturing long-range dependencies in GNNs.

4.2. Comparative analysis of long-range GNN approaches

Our work positions QEL in the broader landscape of approaches for long-range modeling in GNNs:

Table 9 illuminates why QEL offers a favorable trade-off across these dimensions:

- Compared to deep GNNs like GCNII, QEL offers substantially better performance on large-diameter graphs without the over-smoothing issues that typically plague deeper architectures.
- Compared to diffusion-based approaches like APPNP and DIGL, QEL provides more direct optimization of global coherence rather than relying on predefined diffusion processes.
- Compared to Graph Transformers, QEL achieves comparable or better performance with significantly lower computational requirements (5–6× faster training, 3–4× less memory), as detailed in Section 3.4.

5. Limitations and future directions

5.1. Limitations

Despite QEL's strong performance, several limitations remain:

- **Very large graphs**: While our approximation techniques make QEL scalable to graphs with thousands of nodes, extremely large graphs (millions of nodes) would require further optimizations.
- **Heterogeneous graphs**: Our current evaluation focuses on homogeneous graphs. Extending QEL to heterogeneous graphs with multiple node and edge types would broaden its applicability to more complex real-world scenarios.
- **Theoretical guarantees**: While we provide intuitive explanations for QEL's effectiveness, establishing formal theoretical connections between entropy minimization and long-range dependency preservation would strengthen the foundation of our approach.

5.2. Future directions

Several promising opportunities for future research emerge from this work:

- **Hierarchical approaches**: Developing hierarchical or clustered variants of QEL that apply entropy minimization at multiple scales could address scalability challenges for extremely large graphs while maintaining the benefits of global structure preservation.
- **Dynamic graphs**: Investigating how QEL can be adapted for temporal or evolving graphs represents an important research direction, as many real-world networks change over time.
- **Broader applications**: The ability of QEL to effectively model long-range dependencies opens new possibilities for applications where global structure is crucial, including drug discovery (protein-ligand interactions), network analysis (community detection, influence propagation), circuit design (distant component interactions), and computer vision (scene understanding with superpixel graphs). Our strong results on Peptides-struct (37.6% relative improvement), Electronic Circuits (0.066 vs. 0.127 MAE), and PascalVOC-SP/COCO-SP (45.2% and 63.3% improvements) demonstrate significant potential across these diverse domains.
- **Quantum-inspired deep learning**: Beyond specific applications, QEL represents a step toward unifying quantum-inspired techniques with deep learning approaches, potentially opening new research directions at this interdisciplinary interface.

6. Conclusion

In this paper, we introduced the Quantum-Inspired Graph Neural Network (QGNN) with Quantum Entanglement Loss (QEL), a novel approach to address the long-standing challenge of modeling long-range dependencies in graph-structured data. Our comprehensive evaluation on both standard benchmarks and the Long Range Graph Benchmark (LRGB) demonstrates that QEL significantly outperforms existing methods while maintaining practical computational efficiency.

6.1. Summary of contributions

Our primary contributions can be summarized as follows:

- We proposed a novel entropy-based loss function inspired by quantum entanglement principles that explicitly encourages the preservation of long-range dependencies in graph neural networks. On datasets with large diameters (e.g., Peptides-struct), QEL achieves a 37.6% relative improvement over standard GCN and 4.0% over Graph Transformer methods.
- We developed a mathematically rigorous framework for integrating QEL into GNN architectures, with efficient approximation techniques (top-k eigenvalues and power iteration) that reduce computational complexity from $O(n^3)$ to $O(kn^2)$, making QEL practical for large-scale graphs.
- We provided extensive empirical evidence for QEL's effectiveness through hop-stratified analysis, demonstrating that its advantages become more pronounced at larger hop distances. For node pairs 7 + hops apart in PCQM-Contact, QEL achieves 97% better performance than GCN and 18% better than Graph Transformers.

6.2. Practical implications

The practical significance of QGNN with QEL extends beyond academic benchmarks:

- QEL provides a 5–6× speedup and 3–4× memory reduction compared to Graph Transformer approaches while achieving better or comparable performance, making it suitable for resource-constrained environments.
- The model's ability to handle both sparse and dense graphs with varying diameters (from Cora's 6.31 to Peptides' 56.99) demonstrates its versatility across different application domains.

- QEL's improvements on real-world datasets like Electronic Circuits (0.066 MAE vs. 0.127 for GCN) translate to tangible benefits in practical applications.

6.3. Future research directions

Building on this work, we identify several promising directions for future research:

- **Theoretical exploration**: Developing formal connections between von Neumann entropy minimization and spectral graph theory could provide deeper insights into QEL's effectiveness.
- **Architecture extensions**: Integrating QEL with other GNN architectures beyond GCN, such as GAT or GIN, may yield further performance improvements.
- **Application-specific adaptations**: Tailoring QEL for specific domains like drug discovery, knowledge graphs, or recommender systems could address domain-specific challenges.
- **Quantum hardware implementation**: While our approach is designed for classical hardware, exploring potential advantages of implementing QEL on quantum processors represents an intriguing long-term direction.

In conclusion, QGNN with QEL establishes a new paradigm for capturing long-range dependencies in graph neural networks. By drawing inspiration from quantum mechanics to address a fundamental challenge in graph representation learning, our work demonstrates the value of cross-disciplinary approaches in advancing machine learning capabilities. As graph-structured data continues to grow in importance across diverse domains, we believe that entropy-based approaches like QEL will play an increasingly vital role in extracting meaningful insights from complex, interconnected data.

CRediT authorship contribution statement

Muhammad Awais: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Octavian Adrian Postolache:** Writing – review & editing, Visualization, Validation, Resources, Investigation, Funding acquisition; **Sancho Moura Oliveira:** Writing – review & editing, Writing – original draft, Supervision, Resources, Investigation, Funding acquisition, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações, and ManagiDiTH, 101,083,896 DIGITAL-2021-SKILLS-01 EU project.

References

Alon, U., & Yahav, E. (2021). On the bottleneck of graph neural networks and its practical implications. In *International conference on learning representations*.

Balcilar, M., Renton, G., Héroux, P., Gaüzère, B., Adam, S., & Honeine, P. (2021). Analyzing the expressive power of graph neural networks in a spectral perspective. In *International conference on learning representations*.

Banerjee, P. K., Karhadkar, K., Wang, Y. G., Alon, U., & Montúfar, G. (2022). Oversquashing in GNNs through the lens of information contraction and graph expansion. In *58th Annual allerton conference on communication, control, and computing* (pp. 1–8). IEEE.

Chen, M., Wei, Z., Huang, Z., Ding, B., & Li, Y. (2020). Simple and deep graph convolutional networks. *International conference on machine learning*, (pp. 1725–1735).

Dong, Z., Cao, W., Zhang, M., Tao, D., Chen, Y., & Zhang, X. (2023). CktGNN: Circuit graph neural network for electronic design automation. arXiv:2308.16406.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., & Beaini, D. (2022). Long range graph benchmark. In *Advances in neural information processing systems* (pp. 30308–30321). (*vol. 35*).

Gao, Z., Niu, Y., Cheng, J., Tang, J., Li, L., Xu, T., Zhao, P., Tsung, F., & Li, J. (2023). Handling missing data via max-entropy regularized graph autoencoder. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 7651–7659). (*vol. 37*).

Gasteiger, J., Bojchevski, A., & Günnemann, S. (2018). Predict then propagate: Graph neural networks meet personalized pagerank. arXiv:1810.05997.

Gasteiger, J., Weißenberger, S., & Günnemann, S. (2019). Diffusion improves graph learning. *Advances in neural information processing systems*, *32*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

Jin, D., Wang, R., Ge, M., He, D., Li, X., Lin, W., & Zhang, W. (2022). Raw-GNN: Random walk aggregation based graph neural network. arXiv:2206.13953.

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024). A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(1), 18.

Kim, J., Nguyen, D., Min, S., Cho, S., Lee, M., Lee, H., & Hong, S. (2022). Pure transformers are powerful graph learners. In *Advances in neural information processing systems* (pp. 14582–14595). (*vol. 35*).

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., & Tossou, P. (2021). Rethinking graph transformers with spectral attention. In *Advances in neural information processing systems* (pp. 26243–26257). (*vol. 34*).

Lehne, B., & Schlitt, T. (2009). Protein-protein interaction databases: Keeping up with growing interactomes. *Human Genomics*, *3*, 1–7.

Liao, Y., Zhang, X. M., & Ferrie, C. (2024). Graph neural networks on quantum computers. arXiv:2405.17060.

Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., & Ji, S. (2020). Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *22*(6), 3337–3348.

Mialon, G., Chen, D., Selosse, M., & Mairal, J. (2021). Graphit: Encoding graph structure in transformers. In *ICLR Workshop on geometrical and topological representation learning*.

Minello, G., Rossi, L., & Torsello, A. (2019). On the von Neumann entropy of graphs. *Journal of Complex Networks 7*(4), 491–514.

Passerini, F., & Severini, S. (2009). The von Neumann entropy of networks. *International Journal of Agent Technologies and Systems*, *1*(4), 58–67. Also available as arXiv preprint arXiv:0812.2597.

Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., & Yang, B. (2020). Geom-GCN: Geometric graph convolutional networks. In *International conference on learning representations*.

Perrier, E. (2024). Quantum geometric machine learning. arXiv:2409.04955.

Prithviraj, S., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93–106.

Priyadharshini, A. (2024). Quantum-inspired algorithms for AL and machine learning. In *Integration of AI, quantum computing, and semiconductor technology* (p. 79).

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., & Beaini, D. (2022). Recipe for a general, powerful, scalable graph transformer. In *Advances in neural information processing systems* (pp. 14501–14515). (*vol. 35*).

Rong, Y., Huang, W., Xu, T., & Huang, J. (2020). Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations (ICLR)*.

Tönshoff, J., Ritzert, M., Rosenbluth, E., & Grohe, M. (2023a). Where did the gap go? Reassessing the long-range graph benchmark. In *NeurIPS 2023 datasets and benchmarks*.

Tönshoff, J., Ritzert, M., Rosenbluth, E., & Grohe, M. (2023b). Where did the gap go? Reassessing the long-range graph benchmark. arXiv:2309.0036770. citations as of verification.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., & Bronstein, M. M. (2022). Understanding over-squashing and bottlenecks on graphs via curvature. In *International conference on learning representations*.

Ulanov, A. E., Fedorov, I. A., Pushkina, A. A., Kurochkin, Y. V., Ralph, T. C., & Lvovsky, A. I. (2015). Undoing the effect of loss on quantum entanglement. *Nature Photonics*, *9*(11), 764–768.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv:1710.10903.

Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: Methods, applications, and challenges. *Artificial Intelligence Review 56*(7), 6295–6364.

Wu, Q., Zhao, W., Li, Z., Wipf, D. P., & Yan, J. (2022). Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems 35*, 27387–27401.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *International conference on learning representations*.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., & Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning* (pp. 5453–5462). PMLR.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., & Liu, T.-Y. (2021). Do transformers really perform badly for graph representation? In *Advances in neural information processing systems* (pp. 28877–28888). (*vol. 34*).

Zhong, Z., Li, C. T., & Pang, J. (2023). Hierarchical message-passing graph neural networks. *Data Mining and Knowledge Discovery 37*(1), 381–408.