ICTFocus

http://www.ictfocus.org

JOURNAL ARTICLE OPEN ACCESS

Attention Mechanisms in Physics-Inspired Graph Neural Networks for the Max-Cut Problem

Temuujin Delgertsogt[1], Dalaijargal Purevsuren[1,*], and Gantulga Gombojav[1]
[1]Department of Information and Computer Science, School of Information Technology and Electronics, National University of Mongolia, Ulaanbaatar, Mongolia
e-mail: temuujin.d@num.edu.mn, dalaijargal@num.edu.mn, gantulga_g@num.edu.mn
*corresponding author

Abstract

Physics-Inspired Graph Neural Networks (PI-GNNs) reformulate MAX-CUT as QUBO energy minimization, training a GNN to produce soft binary node assignments without labeled data. The baseline PI-GCN uses static, degree-normalized aggregation, while its attention-augmented counterpart PI-GAT—built on GATv2—introduces additional hyperparameters whose effects remain uncharacterized. This paper addresses that gap through controlled experiments on five Gset benchmark graphs (800–10,000 nodes) and five real-world graphs from communication, social, and scientific collaboration networks (500–1,590 nodes). The first experiment examines how attention dropout p_{attn} affects solution quality and convergence stability across 25 seeds per configuration. The second sweeps depth from 1 to 20 layers with 5 independent seeds, quantifying oversmoothing via Dirichlet energy, cosine similarity, and mean absolute deviation. On dense graphs (average degree $\geq$ 10), dropout in $[0.10, 0.20]$ reduces inter-run variance by 30–41% while improving best-cut value; on the sparse graph G70 (average degree$\sim$2), any dropout is detrimental. PI-GAT delays oversmoothing onset by one to two layers versus PI-GCN, at 1.5–1.8$\times$ computational overhead per layer. Real-world validation confirms that depth advantage persists across all five diverse network families, with PI-GAT retaining 25$\times$ more solution quality than PI-GCN at depth $K = 10$ on a communication network (avg. degree 9.61). These findings indicate that graph density and diameter are strong empirical predictors of whether dynamic attention benefits the PI-GNN framework.

Keywords: **Physics-Inspired Graph Neural Networks, Graph Attention Networks, max-cut problem, Combinatorial Optimization, Oversmoothing.**

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

This footnote will be used only by the Editor and Associate Editors. The edition in this area is not permitted to the authors. This footnote must not be removed while editing the manuscript.

1. Introduction

The Maximum Cut (MAX-CUT) problem asks for a partition of the nodes of a weighted undirected graph that maximizes the total weight of edges crossing the partition. It is one of the most extensively studied NP-hard problems in combinatorial optimization [14], with applications in VLSI circuit design, community detection in social networks, and the Ising spin-glass model in statistical physics [3]. Exact solvers are intractable at scale; instead, practitioners rely on approximation algorithms or specialized heuristics. The celebrated 0.878-approximation of Goemans and Williamson [11] establishes the state of the art for polynomial-time algorithms under the Unique Games Conjecture [16], while metaheuristic methods—breakout local search [5], tabu search [21], and GRASP [10]—achieve near-optimal results on structured benchmark instances such as the widely used Gset suite [32].

The rise of deep learning on graphs has motivated a complementary line of work: using graph neural networks (GNNs) to solve combinatorial problems without hand-engineered heuristics [15, 23]. Some use gradient based objective functions to improve performance using massive parallelization offered by GPUs [1]. The works presented in [2, 4, 20, 31] demonstrated that construction heuristics for graph problems can be learned via reinforcement learning. Karalias and Loukas [13] proposed an unsupervised probabilistic framework for QUBO-type objectives. These frameworks are used in quantum approximate optimization algorithms to achieve quantum advantage [6, 9]. Schuetz et al. [27] introduced Physics-Inspired Graph Neural Networks (PI-GNNs), in which a GNN directly minimizes the QUBO energy as a training objective via gradient descent, requiring no labeled solutions. The framework was subsequently extended to graph coloring [28] and to recurrent feature update strategies that improve convergence on larger instances [22].

The central architectural question addressed in this paper concerns the role of attention mechanisms [29] in the PI-GNN framework. The baseline PI-GNN uses GCN-style [17] normalized aggregation (PI-GCN), applying a fixed, degree-based weighting at every layer. A natural refinement is to replace this static aggregation with dynamic, learned attention weights. The original Graph Attention Network (GAT) [30] computes attention scores as a function of pairs of node embeddings, but Brody et al. [7] proved that GAT attention is in fact static: the ranking of a node's neighbors by attention score is independent of the source embedding. Their corrected formulation, GATv2, enables fully dynamic attention by applying a shared nonlinearity to the concatenated source-target embeddings before scoring. Whether this improvement transfers to the unsupervised, energy-minimization setting of PI-GNN is an open empirical question: the training signal is a physics-inspired QUBO objective rather than cross-entropy loss, and the energy landscape contains exponentially many local minima. PI-GAT, the attention-augmented variant we analyze, replaces PI-GCN's static aggregation with multi-head GATv2 attention.

A distinguishing methodological choice of this study is its distributional treatment of results. The original PI-GNN paper [27] reports only the best result over many seeds, a practice that masks pathological variance; practitioners need to understand the full distribution of outcomes. By reporting full distributions over 25 seeds (hyperparameter study) and seed-averaged results with standard deviations (depth study), this paper reveals failure modes that are invisible under best-case reporting—including the finding that PI-GAT without regularization offers no mean-quality advantage over PI-GCN on dense graphs.

Two specific questions motivate this paper. First, how does attention dropout interact with graph structure to determine solution quality and convergence stability? Second, does dynamic attention delay the onset of oversmoothing—the progressive collapse of

node representations at deeper layers [19]—and if so, by how much? Oversmoothing is a fundamental limitation of deeper GNNs [18]: as layers increase, node representations converge toward a common direction, eventually rendering all nodes indistinguishable and degrading cut quality. Answering this question reveals whether PI-GAT enables deeper, and potentially more powerful, architectures. The study is framed as an empirical analysis of attention mechanisms within the PI-GNN framework, with emphasis on understanding the effects of dynamic attention under controlled experimental conditions rather than on benchmarking against state-of-the-art MAX-CUT solvers.

We address both questions through controlled experiments on five Gset benchmark graphs (G14, G15, G22, G49, G70) spanning a wide range of structural regimes, and on five real-world graphs drawn from Network Repository [25]. Our contributions are as follows:

- A systematic study of attention dropout effects in PI-GAT, with 25 seeds per configuration, establishing that the optimal dropout value is governed by graph density and providing actionable configuration guidelines.
- A depth robustness analysis (layers $K = 1$–20) comparing PI-GAT and PI-GCN across three simultaneous oversmoothing metrics, linking the magnitude of attention's advantage to graph diameter.
- A quantification of the computational overhead of dynamic attention and an analysis of the conditions under which the quality gain justifies this cost.
- *A generalization study on five real-world graphs (communication, social, and scientific collaboration networks, 500–1,590 nodes) providing evidence that the framework generalizes across diverse real-world network families.*

The paper is organized as follows. Section 2 presents the problem formulation and attention mechanism. Section 3 describes the experimental setup, presents the hyperparameter sensitivity analysis, and reports the depth robustness and real-world generalization analyses. Section 4 synthesizes the findings and offers practical guidance. Section 5 concludes.

2. Method

2.1. MAX-CUT and the QUBO Formulation

Given an undirected weighted graph $G = (V, E, w)$ with $|V| = n$ nodes, MAX-CUT seeks a partition $(S, V \setminus S)$ maximizing the total crossing edge weight:

$$\text{maximize} \quad C(S) = \sum_{\substack{(i,j)\in E \\ i\in S,\, j\notin S}} w_{ij}. \tag{1}$$

The equivalent Quadratic Unconstrained Binary Optimization (QUBO) formulation is $\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^\top Q\mathbf{x}$, where $Q_{ij} = -\frac{1}{4}w_{ij}$ for $i \neq j$ and $Q_{ii} = \frac{1}{4}\sum_j w_{ij}$. This formulation connects to the Ising Hamiltonian [8] $H(\mathbf{s}) = -\sum_{(i,j)\in E} J_{ij}s_is_j$ via the substitution $s_i = 2x_i - 1$, motivating the "physics-inspired" terminology.

2.2. Physics-Inspired GNNs

Schuetz et al. [27] parameterize $\mathbf{p}(\theta) \in [0,1]^n$ via a GNN and minimize $\mathcal{L}(\theta) = \mathbf{p}(\theta)^\top Q\, \mathbf{p}(\theta)$ using the Adam optimizer, without any labeled solutions. Binary assignments are recovered by thresholding $p_i \geq 0.5$ after training converges. Node features are randomly initialized learnable embeddings, jointly optimized with the GNN parameters.

The baseline PI-GCN uses GCN-style [17] normalized aggregation:

$$\mathbf{h}_i^{(k)} = \sigma\left(\sum_{j \in \tilde{\mathcal{N}}(i)} \frac{1}{\sqrt{\tilde{d}_i \tilde{d}_j}} W^{(k)} \mathbf{h}_j^{(k-1)} \right), \tag{2}$$

where $\tilde{\mathcal{N}}(i)$ includes self-loops, $\tilde{d}_i$ denotes the augmented degree, and $\sigma(\cdot)$ is the ReLU activation function. The aggregation weights $1/\sqrt{\tilde{d}_i \tilde{d}_j}$ are static: they depend only on graph structure, not on the current node embeddings.

2.3. Dynamic Attention: GATv2 and PI-GAT

The original Graph Attention Network (GAT) [30] computes attention scores as $e_{ij} = \text{LeakyReLU}(\mathbf{a}^\top [W\mathbf{h}_i \| W\mathbf{h}_j])$. Brody et al. [7] proved that this formulation is static: for any fixed source node i, the ranking of neighbors by attention score is independent of $\mathbf{h}_i$. GATv2 corrects this limitation by applying a shared linear transformation to the concatenated source-target pair before the nonlinearity:

$$e_{ij}^{(k)} = \mathbf{a}^\top \text{LeakyReLU}\big(W^{(k)} \big[\mathbf{h}_i \,\|\, \mathbf{h}_j\big]\big), \tag{3}$$

so that both $\mathbf{h}_i$ and $\mathbf{h}_j$ interact within the nonlinearity. This makes the attention score—and hence the ranking of neighbors—dependent on the source embedding at every layer, enabling the network to modulate aggregation weights as representations evolve during training.

PI-GAT replaces the static aggregation of PI-GCN with multi-head GATv2 attention:

$$\mathbf{h}_i^{(k)} = \Big\|_{m=1}^{M} \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(k,m)}\, W^{(k,m)} \mathbf{h}_j^{(k-1)} \right), \tag{4}$$

where $\alpha_{ij}^{(k,m)} = \text{softmax}_j(e_{ij}^{(k,m)})$ and $\|$ denotes head concatenation.

All experiments use $M = 2$ attention heads in every hidden layer. The output layer uses a single linear projection followed by a sigmoid activation and does not involve attention aggregation.

2.4. Attention Dropout

Attention dropout p_{attn} stochastically zeros out individual attention coefficients $\alpha_{ij}^{(k,m)}$ after the softmax at each training step. This mechanism is distinct from feature dropout p_{drop}, which masks entries of the embedding $\mathbf{h}_i^{(k)}$ before aggregation. Both act as regularizers but target different components of the computation: attention dropout perturbs the aggregation weights, while feature dropout perturbs the values being aggregated. The two types of dropout are independently controlled in all experiments.

The motivation for studying attention dropout in the PI-GNN setting is as follows. Unlike supervised classification, where the training signal is a dense cross-entropy gradient, the QUBO objective is a global, graph-level energy whose gradient is diffuse. On dense

graphs with many similar-degree neighbors, attention coefficients may settle into unstable regimes—either broadly spread (offering no advantage over PI-GCN) or sharply concentrated on one or two neighbors (sensitive to small embedding perturbations)—producing high variance across random seeds. Attention dropout forces the model to learn aggregation weights that remain effective under random masking, potentially mitigating this fragility.

2.5. Architecture and Training

Both PI-GCN and PI-GAT follow the formulation of Schuetz et al. [27]. The hyperparameter sensitivity experiments use two hidden layers throughout. The initial embedding dimension d_0 is set to the power of two nearest to $\sqrt{n}$: 32 for G14 and G15 ($\sqrt{800} \approx 28$), 64 for G22 and G49, and 128 for G70 ($\sqrt{10{,}000} = 100$). Both architectures employ ReLU activations in hidden layers and a sigmoid output layer that maps the final embedding to $p_i(\theta) \in [0, 1]$. All models are trained with the Adam optimizer for a maximum of 100,000 epochs, with early stopping triggered when the QUBO loss fails to improve for 1,000 consecutive epochs. Each configuration is repeated over 25 independent random seeds to provide reliable estimates of both solution quality and run-to-run variance. Experiments are implemented in PyTorch 2.0 with PyTorch Geometric 2.3 on a single NVIDIA GPU.

3. Experiment

3.1. Oversmoothing and its Metrics

Oversmoothing is the progressive convergence of node representations toward a common direction as network depth increases, first identified by Li et al. [19]. In GCN-style models, each layer multiplies features by the normalized adjacency matrix $\hat{A}$; repeated application drives feature vectors toward the dominant eigenvector of $\hat{A}$, eventually rendering all nodes indistinguishable. Once K exceeds the graph diameter D, the receptive field spans the entire graph at every layer, making full oversmoothing structurally inevitable [26]. For combinatorial optimization, this loss of discriminability directly impairs cut quality.

For attention-based models, the effective propagation range is modulated by learned weights, which may preserve local diversity longer; however, this delay is bounded and the asymptotic behavior remains the same. All three oversmoothing metrics below are computed from the final-layer embeddings $\{\mathbf{h}_i^{(K)}\}$ after training.

Dirichlet energy [19] measures local edge-wise feature variation:

$$E^{(K)} = \frac{1}{|E|} \sum_{(i,j)\in E} \|\mathbf{h}_i^{(K)} - \mathbf{h}_j^{(K)}\|^2. \tag{5}$$

As adjacent embeddings become identical, $E^{(K)} \to 0$.

Global cosine similarity measures directional collapse across connected pairs:

$$\mathrm{Cos}^{(K)} = \frac{1}{|E|} \sum_{(i,j)\in E} \cos\left(\mathbf{h}_i^{(K)}, \mathbf{h}_j^{(K)}\right). \tag{6}$$

Full collapse corresponds to $\mathrm{Cos}^{(K)} \to 1$.

Mean Absolute Deviation (MAD) measures the average cosine dissimilarity from the mean representation:

$$\mathrm{MAD}^{(K)} = \frac{1}{|V|} \sum_{i\in V} \left(1 - \frac{\mathbf{h}_i^{(K)\top} \bar{\mathbf{h}}^{(K)}}{\|\mathbf{h}_i^{(K)}\| \, \|\bar{\mathbf{h}}^{(K)}\|}\right). \tag{7}$$

TABLE I
Gset Benchmark Properties and Best-Known MAX-CUT Values

G	$\|V\|$	$\|E\|$	$\bar{d}$	Diam.	Density	Best Cut
G14	800	4,694	11.74	6	0.0147	3,064[a]
G15	800	4,661	11.65	6	0.0146	3,050[a]
G22	2,000	19,990	19.99	4	0.0100	13,359[b]
G49	3,000	6,000	4.00	65	0.0013	6,000[c]
G70	10,000	9,999	2.00	29[1]	0.0002	9,541[d]

[a] [5]; [b] [10]; [c] [12]; [d] [21];

As all representations converge toward a single direction, MAD $\to 0$.

Because the Dirichlet energy for the Gset benchmarks is computed on raw embeddings rather than L_2-normalized embeddings, a massive inflation in vector magnitudes at extreme depths masks representational collapse, causing the energy value to rise even as the representations lose discriminative capacity. In contrast, we have applied L_2-normalization to the node embeddings for our extended real-world graph experiments, with the corresponding metric visualizations provided in the Appendix. Employing multiple metrics remains essential because each captures a different facet of collapse: unnormalized Dirichlet energy reflects local edge-wise divergence scaled by magnitude, while Cosine Similarity and MAD reflect directional alignment and global spread, respectively.

3.2. Benchmark Graphs

Five graphs from the Gset benchmark suite [32] are selected to span contrasting structural regimes. Table I summarizes their key properties. G14 and G15 are small and moderately dense, with diameter 6. G22 is larger and highly dense—average degree 20, diameter 4—making it the most susceptible to oversmoothing. G49 has regular degree 4 and is significantly larger and sparser. G70 is the largest, with near-tree sparsity (average degree 2) and largest-component diameter 29, representing the structural regime most amenable to deep network exploration.

3.3. Experimental Designs

Hyperparameter sensitivity: For each graph, feature dropout p_{drop} and learning rate η are fixed at their per-graph optimized values (Table II), and attention dropout $p_{\text{attn}} \in \{0.00, 0.05, 0.10, 0.20\}$ is varied for PI-GAT. The PI-GCN baseline, which lacks an attention dropout parameter, is evaluated under the same p_{drop} and η. This design isolates the effect of attention dropout from all other hyperparameters.

Depth robustness: Network depth is swept from $K = 1$ to $K = 20$ hidden layers on G14, G22, and G70, which together represent moderately dense, highly dense with small diameter, and sparse with large diameter structural regimes, respectively. To isolate depth effects from model capacity, the hidden dimension is fixed at 32 for all graphs and both architectures. PI-GAT is evaluated with $p_{\text{attn}} = 0$ so that the only architectural difference from PI-GCN is the aggregation function, enabling a controlled depth comparison. The three oversmoothing metrics are recorded from final-layer representations after training at each depth. The Gset depth sweeps use a single seed per configuration; the depth

[1] G70 is disconnected; the value is the diameter of its largest connected component.

TABLE II
Fixed Hyperparameters for the Sensitivity Experiments (Per Graph)

G	p^*_{drop}	η^*	PI-GCN Best Cut	PI-GCN Std
G14	0.2	0.004	2,905	261
G15	0.1	0.002	2,919	169
G22	0.2	0.003	13,039	25
G49	0.4	0.005	5,506	8
G70	0.4	0.004	9,157	24

TABLE III
PI-GAT MAX-CUT Results by Attention Dropout Value (Best / Mean / Std over 25 Seeds; PI-GCN Baseline Shown for Comparison). Bold marks the best value in each metric group per row.

G	**PI-GAT** (p_{attn}=0.00)			**PI-GAT** (p_{attn}=0.05)			**PI-GAT** (p_{attn}=0.10)			**PI-GAT** (p_{attn}=0.20)			**PI-GCN (baseline)**		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
G14	2,924	2,576	253	2,933	2,781	114	2,939	2,799	113	**2,940**	**2,811**	150	2,905	2,555	261
G15	2,868	2,533	265	2,907	2,785	112	2,912	**2,811**	89	**2,937**	2,802	122	2,919	2,664	169
G22	12,630	10,815	1,701	12,860	11,384	1,814	12,916	11,944	1,184	12,814	11,714	787	**13,039**	**12,994**	25
G49	5,680	5,386	248	5,694	5,497	178	**5,742**	5,558	97	5,702	**5,575**	66	5,506	5,489	8
G70	**9,210**	9,004	212	9,137	9,002	124	9,032	8,917	95	8,746	8,628	78	9,157	**9,129**	24

study is repeated over 5 independent seeds per configuration for the real-world graph generalization experiment described in Section 3.6.

Experimental scope and comparative design: While this study focuses on comparing PI-GCN and PI-GAT within the Physics-Inspired Graph Neural Network (PI-GNN) framework, it is useful to place them in a broader context. Classical methods such as Simulated Annealing use stochastic local search for Max-Cut, while learning-based approaches employ Graph Neural Networks or Reinforcement Learning to learn solution strategies. In contrast, PI-GNN casts Max-Cut as a QUBO energy minimization problem and learns soft binary assignments under this formulation. Within this framework, the comparison between PI-GCN and PI-GAT is a controlled architectural study: PI-GAT extends PI-GCN by introducing attention-based aggregation while preserving the same objective, training procedure, and energy formulation, allowing us to isolate the effect of attention on solution quality, convergence, and oversmoothing without confounding factors. Broader comparisons with heterogeneous solvers are left for future work. Accordingly, this paper is an empirical analysis of attention mechanisms within the PI-GNN framework and is not intended as a benchmark of state-of-the-art MAX-CUT solvers.

3.4. Hyperparameter Sensitivity Analysis

3.4.1 Main Results

Table III presents the complete results of the sensitivity experiments: best cut value, mean cut value, and standard deviation over 25 seeds for each PI-GAT attention dropout value, alongside the PI-GCN baseline. Bold entries indicate the best value in each column group per graph.

3.4.2 Dense Graphs: Dropout Stabilizes and Improves Quality

On G14 and G15 (average degree ≈ 12, diameter 6), the strongest tested dropout ($p_{\text{attn}} = 0.20$) yields the best overall performance. For G14, it achieves the highest best cut (2,940) and mean (2,811), while reducing the standard deviation from 253 to 150—a 41% reduction in run-to-run variance compared to no dropout. G15 exhibits the same pattern: best cut 2,937, with variance approximately halved.

The variance reductions are statistically significant: Levene's test comparing the cut distributions at $p_{\text{attn}} = 0$ versus $p_{\text{attn}} = 0.10$ yields $W = 4.8$, $p = 0.033$ for G14 and $W = 5.18$, $p = 0.0274$ for G15, confirming that the stabilization reflects a genuine change in learning dynamics rather than a sampling artifact.

More revealingly, without any attention dropout ($p_{\text{attn}} = 0$), PI-GAT on G14 achieves a mean of only 2,576—effectively equivalent to PI-GCN's mean of 2,555. Dynamic attention without regularization provides no advantage on a dense graph; it merely adds instability. This finding would be invisible if one reported only best-case results.

G22 (diameter 4, average degree 20) is the densest graph in the benchmark set. Here, moderate dropout $p_{\text{attn}} = 0.10$ achieves the best PI-GAT cut (12,916) and reduces variance from 1,701 (at $p_{\text{attn}} = 0$) to 1,184—a 30% reduction. Increasing to $p_{\text{attn}} = 0.20$ further reduces variance to 787 but lowers the best cut to 12,814, revealing a trade-off between reliability and peak performance.

The sparse-regular graph G49 (average degree 4) exhibits a consistent but more moderate benefit from dropout. The setting $p_{\text{attn}} = 0.10$ yields the best cut (5,742) and $p_{\text{attn}} = 0.20$ yields the best mean (5,575), both substantially exceeding no dropout (5,680 best, 5,386 mean) and surpassing PI-GCN (5,506). The gains represent genuine quality improvements, not merely variance reduction.

3.4.3 Sparse Graphs: Dropout Is Counterproductive

G70 presents the opposite pattern. At $p_{\text{attn}} = 0$, PI-GAT achieves its best cut (9,210) and highest mean (9,004). Adding any dropout degrades both metrics monotonically: by $p_{\text{attn}} = 0.20$, the best cut falls to 8,746 and the mean to 8,628—a 5.0% best-cut degradation relative to the unregularized configuration. PI-GCN on G70 achieves best 9,157 and mean 9,129 with a standard deviation of just 24, confirming that this near-tree structure supports stable uniform aggregation.

The contrast between G70 and the dense graphs is stark. The same architecture that struggles without dropout on dense graphs performs optimally without dropout on G70, and adding dropout only causes degradation. As discussed in Section 2.4, on G70 (average degree 2), the natural degree heterogeneity produces stable attention signals without external regularization; dropout serves only to disrupt an already informative signal.

3.4.4 Comparison with PI-GCN

Comparing per-graph optimal PI-GAT configurations against PI-GCN reveals a nuanced picture. PI-GAT at its best configuration outperforms PI-GCN in best-cut value on G14, G15, and G49. The most pronounced gain occurs on G49, where PI-GAT (5,742) exceeds PI-GCN (5,506) by 236 cut edges—a 4.3% improvement. On G22 and G70, however, PI-GCN's substantially lower variance makes it the more reliable choice for practitioners who cannot afford many independent runs. The conclusion is not that one architecture dominates universally, but that the optimal choice depends on graph structure and operational constraints.

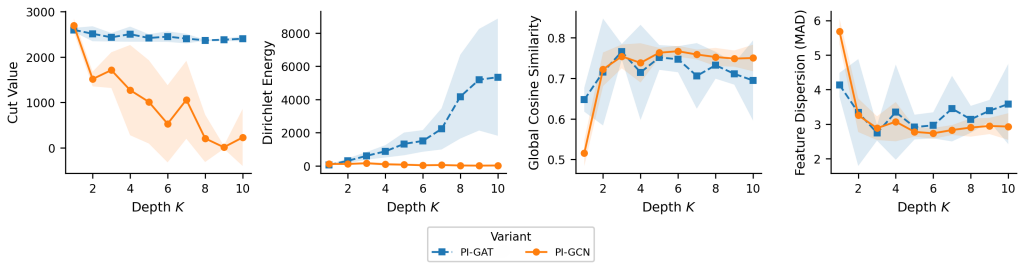


Figure 1. Cut value and oversmoothing metrics as a function of network depth K for PI-GATv2 and PI-GCN on G14 (diameter 6, $\bar{d} = 11.74$). From left to right: cut value, Dirichlet energy, global cosine similarity, and feature dispersion (MAD). PI-GATv2 maintains solution quality and representational diversity at greater depth than PI-GCN, with oversmoothing onset delayed by approximately two layers.

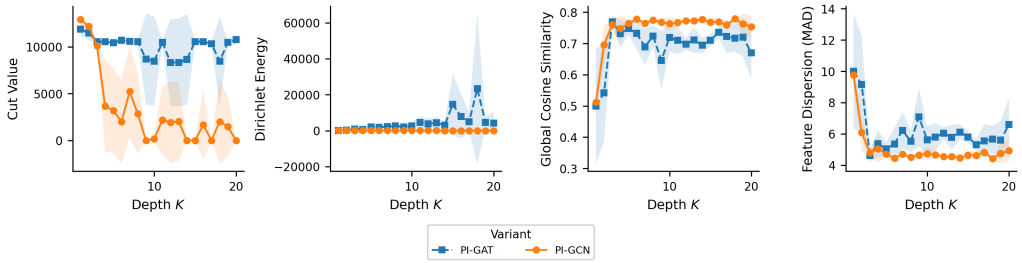


Figure 2. Cut value and oversmoothing metrics vs. depth K on G22 (diameter 4, $\bar{d} = 19.99$). The highly dense, small-diameter structure causes earlier oversmoothing onset for both architectures compared to G14, with PI-GATv2 providing a single-layer advantage over PI-GCN.

TABLE IV
Oversmoothing Onset: First Depth K at Which Cut Value Falls More Than 5% Below Peak (PI-GCN vs. PI-GAT)

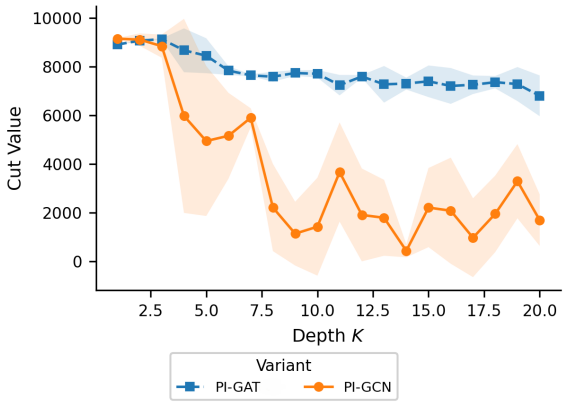
G	Diam.	$\bar{d}$	Onset depth		Δ
			PI-GCN	PI-GAT	
G14	6	11.74	$K \geq 3$	$K \geq 5$	+2 layers
G22	4	19.99	$K \geq 3$	$K \geq 4$	+1 layer
G70	29	2.00	$K \geq 2$	$K \geq 4$	+2 layers

3.5. Depth Robustness and Oversmoothing

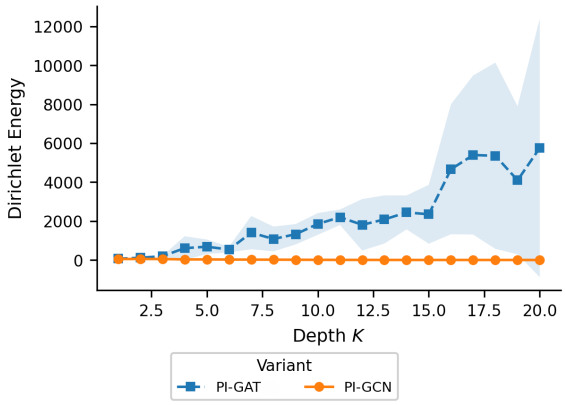
3.5.1 Cut Quality vs. Depth

Figs. 1–3d show how solution quality and oversmoothing metrics evolve as a function of the number of hidden layers K for both architectures on the three analysis graphs. Table IV summarizes oversmoothing onset, defined as the smallest K at which cut value falls more than 5% below its peak.

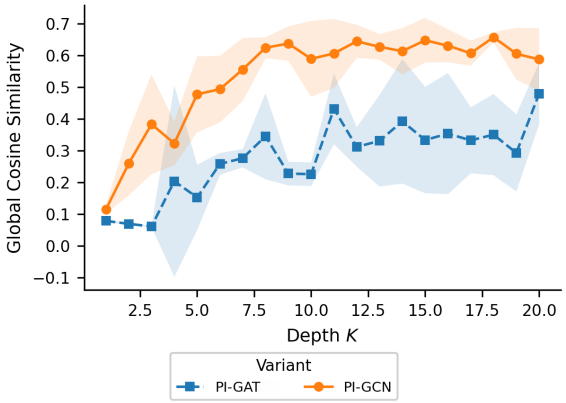
The depth-quality curves reveal a pattern consistent with predictions from oversmoothing theory. On G70 (largest-component diameter 29), PI-GCN degrades sharply at $K = 2$, dropping from a peak near 9,143 at $K = 1$ to substantially degraded quality by $K = 3$. PI-GAT sustains peak quality through $K = 3$ before declining, providing a two-layer depth advantage (Fig. 3a). This is the largest gain observed across the three



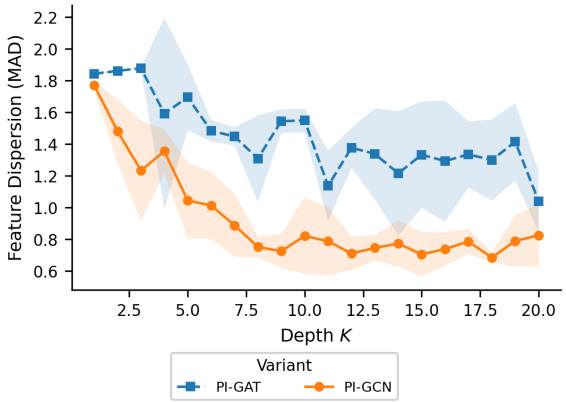
(a) Cut value vs. depth. PI-GATv2 sustains near-peak quality through $K = 3$–4, while PI-GCN degrades sharply beyond $K = 2$.



(b) Dirichlet energy vs. depth. PI-GCN collapses to near zero (local feature uniformity); PI-GATv2 maintains increasing edge-wise variation.



(c) Global cosine similarity vs. depth. PI-GCN rises rapidly toward 0.6–0.65; PI-GATv2 stays significantly lower across all depths.



(d) Mean Absolute Deviation (MAD) vs. depth. PI-GATv2 maintains higher feature dispersion; PI-GCN drops rapidly and plateaus lower.

Figure 3. Cut value and three oversmoothing metrics vs. depth on G70 (largest-component diameter 29). All three diagnostics yield concordant evidence that PI-GATv2 delays representational collapse on this large-diameter sparse graph.

graphs, consistent with the large diameter: the graph contains many hops over which dynamic attention can preserve structural diversity before the receptive field spans the entire graph.

On G22 (diameter 4), oversmoothing onset occurs earlier for both architectures (Fig. 2). PI-GCN begins to degrade at $K = 3$ and PI-GAT at $K = 4$—a single-layer advantage. This smaller margin is expected: with diameter 4, any aggregation scheme propagates information across the full graph within a few layers, leaving limited room for attention to delay the inevitable convergence. On G14 (diameter 6), PI-GAT maintains quality through $K = 4$ while PI-GCN degrades at $K = 3$, yielding the same two-layer advantage as G70 (Fig. 1).

3.5.2 Oversmoothing Metrics

Figs. 3a–3d show the cut value and all three oversmoothing metrics as a function of depth on G70, where the contrast between architectures is most pronounced.

Table V provides representative numerical values at selected depths for G22.

The collapse trajectory for PI-GCN on G22 is instructive. By $K = 4$, cosine similarity

TABLE V
Oversmoothing Metrics at Selected Depths on G22

	PI-GCN				PI-GAT			
K	Cut	Dir.E	Cos	MAD	Cut	Dir.E	Cos	MAD
2	12,882	1.6×10^{4}	0.31	0.32	12,916	2.1×10^{4}	0.19	0.41
4	9,795	2.3×10^{7}	1.00	0.00	11,430	4.8×10^{5}	0.47	0.21
6	9,755	4.3×10^{9}	1.00	0.00	10,890	3.2×10^{6}	0.82	0.07
10	9,743	6.1×10^{11}	1.00	0.00	10,511	2.9×10^{8}	0.98	0.01

TABLE VI
Mean Training Time Per Hidden Layer: PI-GCN vs. PI-GAT

G	PI-GCN (s/layer)	PI-GAT (s/layer)	Ratio
G14	4.3	7.8	1.8×
G22	7.5	11.1	1.5×
G70	21.2	32.8	1.5×

reaches 1.00 and MAD collapses to zero, indicating full representational collapse. Dirichlet energy simultaneously rises to 2.3×10^{7}: because the standard formula (Eq. 5) uses unnormalized embeddings, the growth in embedding magnitude from successive ReLU and weight-matrix operations inflates the squared-difference term even as directions converge. On ℓ_2-normalized embeddings the metric instead falls from $\approx$0.62 at $K=2$ toward zero by $K=4$, consistent with cosine similarity and MAD. We retain the unnormalized formulation for comparability with prior work [19, 26], treating the normalized cosine similarity and MAD as the authoritative collapse signals.

Despite representational collapse, PI-GCN still produces a cut of approximately 9,795 at $K=4$, because the binary threshold on the output sigmoid can still partition nodes into two groups. However, the gap relative to the healthy $K=2$ result (12,882) reveals the true cost: over 3,000 cut edges lost.

PI-GAT at the same depth retains cosine similarity of 0.47 and MAD of 0.21—far from complete collapse—and achieves a cut of 11,430, some 1,635 edges better than PI-GCN. By $K=10$, PI-GAT approaches collapse itself (cosine 0.98), but the delay spans two to four layers throughout the measured range.

The consistency across all three metrics merits emphasis. Dirichlet energy, cosine similarity, and MAD capture different aspects of representational collapse—local edge-wise differences, global directional alignment, and spread around the mean, respectively—yet they yield concordant conclusions at every depth point.

3.5.3 Computational Overhead

Table VI reports the mean training time per hidden layer on the three analysis graphs.

The overhead is 1.5–1.8× per layer, arising from attention score computation (one matrix multiplication per edge per head), softmax normalization over each node's neighborhood, and subsequent weighted aggregation. All operations scale with the number of edges rather than node pairs, bounding the overhead independently of graph size. On G14 and G49, where PI-GAT at its optimal dropout outperforms PI-GCN in best-cut

TABLE VII
Real-World Benchmark Graph Properties

Graph	**Domain**	$\|V\|$	$\|E\|$	$\bar{d}$	Density	Conn.
email	Communication	1,134	5,451	9.61	0.0085	No
Harvard500	Social (FB100)	500	2,043	8.17	0.0164	Yes
homer	Co-occurrence	561	1,629	5.81	0.0104	No
H	Collaboration	706	1,392	3.94	0.0056	Yes
netscience	Co-authorship	1,590	2,742	3.45	0.0022	No

TABLE VIII
Depth Robustness on Real-World Graphs (5 Seeds per Depth). Onset is the first K where mean cut falls more than 5% below peak. Cut@10 is the mean cut value at $K = 10$.

Graph	$\bar{d}$	**PI-GCN**			**PI-GAT**			Δ
		Best	Cut@10	Onset	Best	Cut@10	Onset	
email	9.61	3,597	118	$K \geq 2$	3,658	3,052	$K \geq 3$	+1
Harvard500	8.17	1,239	186	$K \geq 3$	1,455	1,315	$K \geq 6$	+3
homer	5.81	1,045	124	$K \geq 2$	1,090	944	$K \geq 3$	+1
H	3.94	1,237	163	$K \geq 4$	1,237	1,095	$K \geq 5$	+1
netscience	3.45	1,512	187	$K \geq 2$	1,812	1,173	$K \geq 3$	+1

value, this overhead is a reasonable price. On G22 and G70, where PI-GCN matches or exceeds PI-GAT in mean quality with far lower variance, the additional computational cost is not warranted.

3.6. Generalization to Real-World Graph Families

To assess whether the density-diameter framework established on Gset generalizes beyond synthetic benchmarks, we apply both architectures to five real-world Network Repository [25]. Table VII summarizes graph properties. These graphs span average degrees from 3.45 (netscience, scientific coauthorship) to 9.61 (email-EuAll, communication network), and include both connected and multi-component instances. For each graph, the depth sweep $K = 1$–20 is repeated over 5 independent seeds per configuration, following the same protocol as the Gset depth experiments.

Table VIII presents peak cut values, best cuts at depth $K = 10$, and oversmoothing onset for both architectures on all five graphs.

The real-world results confirm the density-diameter pattern with notable consistency across all five graphs. On the two denser instances (email, Harvard500), PI-GCN degrades catastrophically beyond $K = 2$–3, while PI-GAT retains substantial quality to considerably greater depth. The most pronounced illustration is the email graph (average degree 9.61, structurally similar to Gset G14/G15): at depth $K = 10$, PI-GAT achieves a mean cut of 3,052 versus PI-GCN's mean of only 118—a 25-fold retention of solution quality. On Harvard500 (social network, average degree 8.17), PI-GAT's best cut (1,455) exceeds PI-GCN's (1,239) by 17.4%, and depth advantage extends to $K \geq 6$.

On the sparser graphs (homer, H, netscience), the pattern follows the Gset moderate-density regime. The H graph (average degree 3.94, structurally similar to G49) shows near-parity in best cut with onset delayed by one layer. The netscience graph (average degree 3.45, scientific coauthorship network with 397 connected components) displays a large absolute quality gain at all depths: PI-GAT best cut 1,812 versus PI-GCN's 1,512, a 19.8% improvement driven by structural heterogeneity across its many components providing diverse attention signals even without dropout. At $K = 10$, PI-GAT still achieves mean cut 1,173 versus PI-GCN's 187—a 6.3-fold advantage.

Together, these results indicate that graph density and diameter appear to be important structural predictors of dynamic attention advantage across both synthetic benchmark and real-world network families.

4. Discussion

4.1. A Unified Account

The hyperparameter sensitivity and depth robustness results converge on a single underlying principle: the benefit of dynamic attention in PI-GAT is contingent on whether the graph's local structure provides informative, stable signals for the attention mechanism to exploit.

On dense graphs (G14, G15, G22), structural homogeneity renders attention coefficients unstable without external regularization, manifesting directly as high run-to-run variance. The same structural homogeneity limits the duration over which attention can delay oversmoothing: small diameters (4–6) ensure rapid global information propagation regardless of aggregation weights, bounding the depth advantage at one to two layers.

On the sparse near-tree G70, the converse holds. Structural heterogeneity—degree variation and long paths—produces naturally stable attention signals without dropout, explaining why $p_{\text{attn}} = 0$ is optimal. The large diameter (29) provides ample room for attention to delay oversmoothing, yielding the largest depth advantage (+2 layers).

The mechanism relates to the structure of the attention softmax. On dense graphs with many similar-degree neighbors, attention coefficients tend to settle into unstable regimes: either broadly spread (uniform-like, offering no advantage over PI-GCN) or sharply concentrated on one or two neighbors (over-committed, sensitive to small embedding perturbations). Attention dropout mitigates this fragility by forcing the model to learn aggregation weights that remain effective even when individual connections are randomly masked. On G70 (average degree 2), natural degree heterogeneity produces informative attention without external regularization.

The practical implication is that whether PI-GAT should be preferred over PI-GCN, and at what dropout level, can be estimated from two easily computed graph statistics—average degree and diameter—before running any experiments.

4.2. Limitations

This study covers the MAX-CUT problem exclusively. The Gset hyperparameter sensitivity analysis uses a fixed attention dropout grid $\{0.00, 0.05, 0.10, 0.20\}$; finer-grained schedules and interactions with learning rate are not explored. The Gset depth study uses a single seed per configuration; variance in the depth-quality trajectory is not characterized for those graphs (the real-world depth experiments in Section 3.6 use 5 seeds per depth, partially addressing this limitation). The real-world graph depth experiments use 5 seeds rather than 25, limiting statistical power for variance comparisons on those instances. The interaction between depth and attention dropout is not characterized;

optimal depth-aware dropout schedules remain an open question. The number of attention heads ($M = 2$) is fixed throughout. The real-world graphs extend only to average degree 3.45, so the near-tree regime (average degree $\sim$2) is evaluated solely on the synthetic graph G70; whether the observed patterns hold for sparse real-world networks remains untested. Finally, potential oversmoothing remedies such as residual connections, PairNorm [26], and DropEdge [24] are not evaluated.

5. Conclusion

This paper investigated two aspects of Physics-Inspired Graph Attention Networks for MAX-CUT that prior work had left uncharacterized: the effect of attention dropout on solution quality and convergence stability, and the extent to which dynamic attention delays oversmoothing in deeper architectures.

The hyperparameter sensitivity analysis demonstrated that attention dropout is not a uniformly beneficial regularizer. On dense and moderate-density graphs (G14, G15, G22, G49), $p_{\text{attn}} \in [0.10, 0.20]$ reduces inter-run standard deviation by 30–41% while improving best-cut value; without dropout, PI-GAT offers no advantage over the static PI-GCN baseline. On the very sparse G70 (average degree 2, largest-component diameter 29), any dropout is detrimental because dynamic attention already produces stable, informative weights from the graph's structural heterogeneity. The principal insight is that the appropriate dropout setting can be predicted from graph density and diameter before experiments are conducted.

The depth robustness analysis demonstrated that PI-GAT consistently delays oversmoothing onset by one to two layers compared to PI-GCN, confirmed by all three metrics (Dirichlet energy, cosine similarity, MAD). The magnitude of the delay is governed by graph diameter: larger diameters afford attention greater capacity to preserve local diversity. The computational cost of dynamic attention is 1.5–1.8$\times$ per layer, a reasonable trade-off on graphs where attention provides a quality benefit. Real-world validation on five diverse network families confirms that the density-diameter framework generalizes beyond synthetic benchmarks: on a communication network (email, average degree 9.61), PI-GAT retains 25$\times$ more solution quality than PI-GCN at depth $K = 10$, and on a social network (Harvard500), PI-GAT's best cut exceeds PI-GCN's by 17.4% with depth advantage extending to $K \geq 6$.

Together, these results provide a principled basis for architecture selection in PI-GNN-based combinatorial optimization: graph density and diameter appear to be important structural predictors of whether and how to deploy dynamic attention. Future directions include investigating the interaction between attention head count and depth robustness, evaluating depth-enabling mechanisms such as residual connections within the PI-GAT framework, and extending systematic comparison against classical solvers.

REFERENCES

[1] Ismail Alkhouri, Mian Wu, Cunxi Yu, Jia Liu, Rongrong Wang, and Alvaro Velasquez. A scalable lift-and-project differentiable approach for the maximum cut problem. *arXiv preprint arXiv:2509.18612*, 2025.

[2] Paul Almasan, José Suárez-Varela, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. *Computer Communications*, 196:184–194, 2022.

[3] Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988.

[4] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[5] Una Benlic and Jin-Kao Hao. Breakout local search for the max-cut problem. *Engineering Applications of Artificial Intelligence*, 26(3):1162–1173, 2013.

[6] Bikrant Bhattacharyya, Michael Capriotti, and Reuben Tate. Solving general qubos with warm-start qaoa via a reduction to max-cut. *arXiv preprint arXiv:2504.06253*, 2025.

[7] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[8] Barry A Cipra. An introduction to the ising model. *The American Mathematical Monthly*, 94(10):937–959, 1987.

[9] Jose Falla, Quinn Langfitt, Yuri Alexeev, and Ilya Safro. Graph representation learning for parameter transferability in quantum approximate optimization algorithm. *Quantum Machine Intelligence*, 6(2):46, 2024.

[10] Paola Festa, Panos M. Pardalos, Mauricio G. C. Resende, and Celso C. Ribeiro. Randomized heuristics for the max-cut problem. *Optimization Methods and Software*, 17(6):1033–1058, 2002.

[11] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[12] Christoph Helmberg and Franz Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[13] Dimitris Karalias and Andreas Loukas. Erdos goes neural: An unsupervised learning framework for combinatorial optimization on graphs. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, pages 6659–6672, 2020.

[14] Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum, New York, NY, USA, 1972.

[15] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, pages 6348–6358, 2017.

[16] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.

[18] Jin Li, Qirong Zhang, Wenxi Liu, Antoni B Chan, and Yang-Geng Fu. Another perspective of over-smoothing: Alleviating semantic over-smoothing in deep gnns. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6897–6910, 2024.

[19] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised classification. In *Proc. AAAI Conf. Artif. Intell.*, volume 32, pages 3538–3545, 2018.

[20] Gabriel Maliakal, Ismail Alkhouri, Alvaro Velasquez, Adam M Alessio, and Saiprasad Ravishankar. A dataless reinforcement learning approach to rounding hyperplane optimization for max-cut. *arXiv preprint arXiv:2505.13405*, 2025.

[21] Gintaras Palubeckis. Multistart tabu search strategies for the unconstrained binary quadratic optimization problem. *Annals of Operations Research*, 131:259–282, 2004.

[22] Daria Pugacheva, Andrey Ermakov, Ivan Lyskov, Ilya Makarov, and Yuri Zotov. Enhancing GNNs performance on combinatorial optimization by recurrent feature update. *arXiv preprint arXiv:2407.16468*, 2024.

[23] Yeqing Qiu, Ye Xue, Akang Wang, Yiheng Wang, Qingjiang Shi, and Zhi-Quan Luo. Ros: A gnn-based relax-optimize-and-sample framework for max-k-cut problems. *arXiv preprint arXiv:2412.05146*, 2024.

[24] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[25] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[26] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

[27] Martin J. A. Schuetz, J. Kyle Brubaker, and Helmut G. Katzgraber. Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence*, 4:367–377, 2022.

[28] Martin J. A. Schuetz, J. Kyle Brubaker, Zhuo Zhu, and Helmut G. Katzgraber. Graph coloring with physics-inspired graph neural networks. *Physical Review Research*, 4(4):043131, 2022.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018.

[31] Yang Wang and Xiaoxiang Liang. Application of reinforcement learning methods combining graph neural networks and self-attention mechanisms in supply chain route optimization. *Sensors*, 25(3):955, 2025.

[32] Yinyu Ye. The Gset dataset. [Online]. Available: `https://web.stanford.edu/~yyye/yyye/Gset/`, 2003.

ACKNOWLEDGEMENTS

The research has received funding from the National University of Mongolia under grant agreement PROF2024-3113, PROF2026-3363. The authors would like to express their sincere gratitude to the National University of Mongolia for its financial support and for providing an excellent research environment that made this work possible.

BIOGRAPHIES

Temuujin Delgertsogt received the B.Sc. degree in Computer Science from the National University of Mongolia (NUM), Ulaanbaatar, Mongolia, in 2024, and the M.Sc. degree in Computer Science from the same institution in 2026, under the supervision of Dr. Dalaijargal Purevsuren. His research interests include physics-inspired graph neural networks, and the analysis of networked systems.

Dalaijargal Purevsuren received his B.Sc. (2009) and M.Sc. (2011) degrees in Computer Science from the National University of Mongolia, and his Ph.D. in Computer Science from Harbin Institute of Technology in 2017. His research interests include randomized algorithms, graph mining, and the analysis of networked systems.

Gantulga Gombojav obtained B.S. computer science and M.S. Information technology from University of Madras, India and National University of Mongolia respectively in 2012 and 2016. Currently, he is a Ph.D. student at the National University of Mongolia. His research interests include network analysis and algorithm design.

A. Metric Visualizations for Real-World Graph Extensions

To isolate the embedding magnitude inflation artifact observed in the unnormalized Gset experiments, this section provides a metric-by-metric breakdown of the structural dynamics and optimization performance across all five extended real-world topologies. All structural metrics are computed using L_2-normalized node embeddings, and all values are averaged over 5 independent random initialization seeds across depths $K = 1$ to $K = 20$.

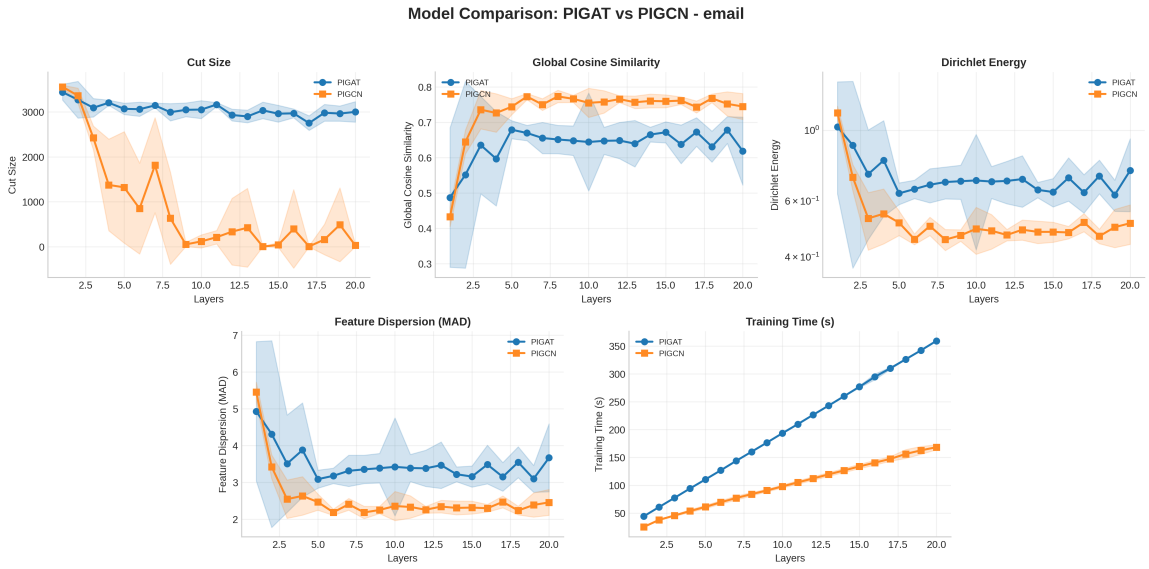


Figure 4. Discrete Max-Cut value optimization curves for the Email network.

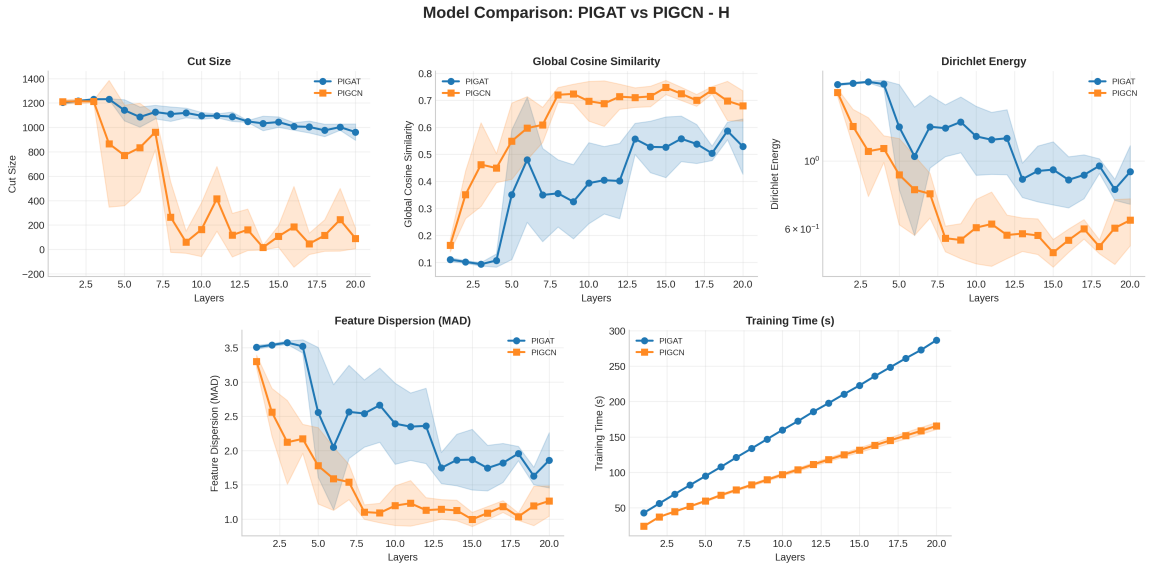


Figure 5. Discrete Max-Cut value optimization curves for the H Pylori network.

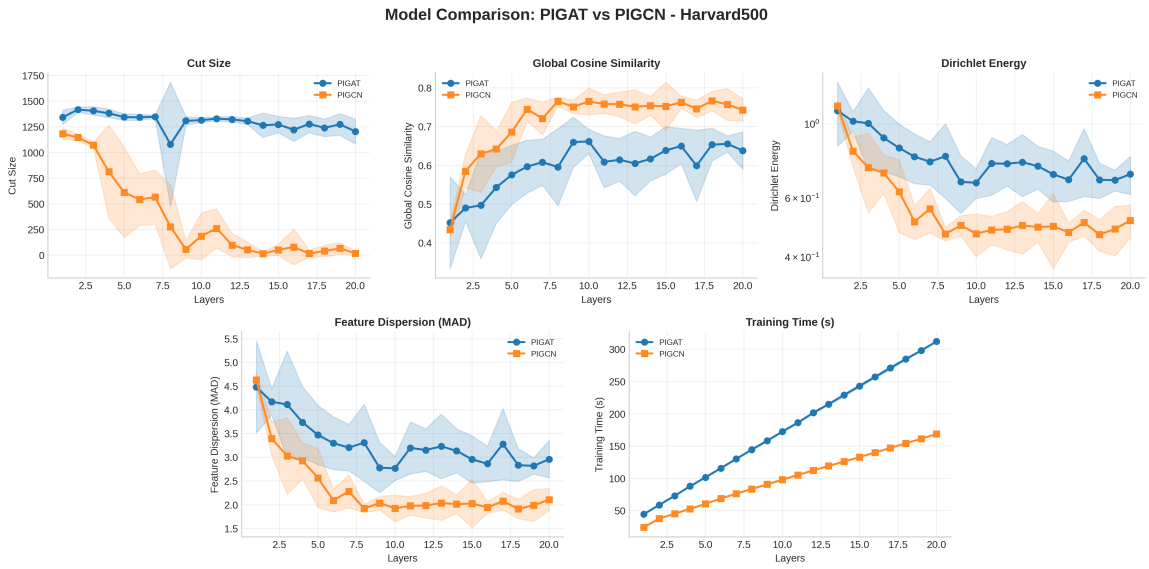


Figure 6. Discrete Max-Cut value optimization curves for the Harvard500 network.

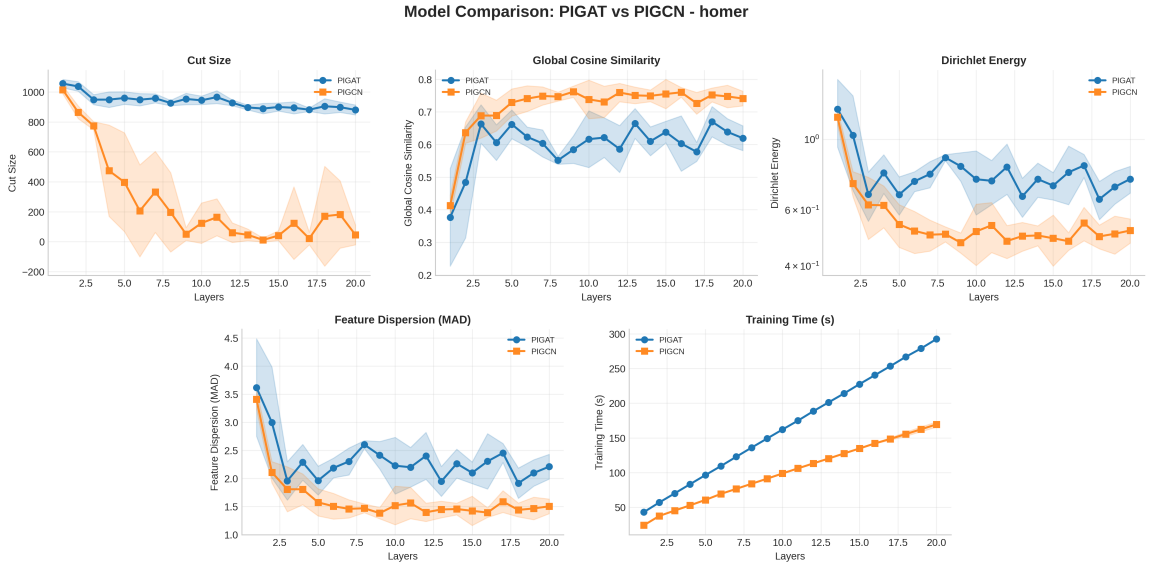


Figure 7. Discrete Max-Cut value optimization curves for the Homer network.

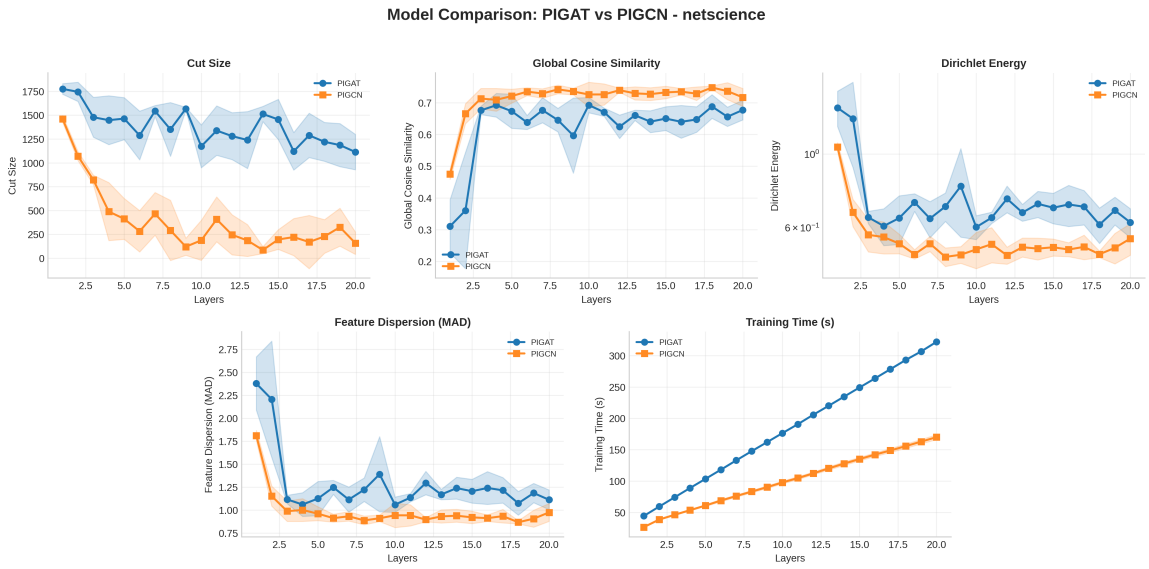


Figure 8. Discrete Max-Cut value optimization curves for the NetScience network.