HIGH-DEGREE NODE ENERGY SMEARING FOR GRAPH NEURAL NETWORKS QUANTIZATION

Anonymous authors
Paper under double-blind review

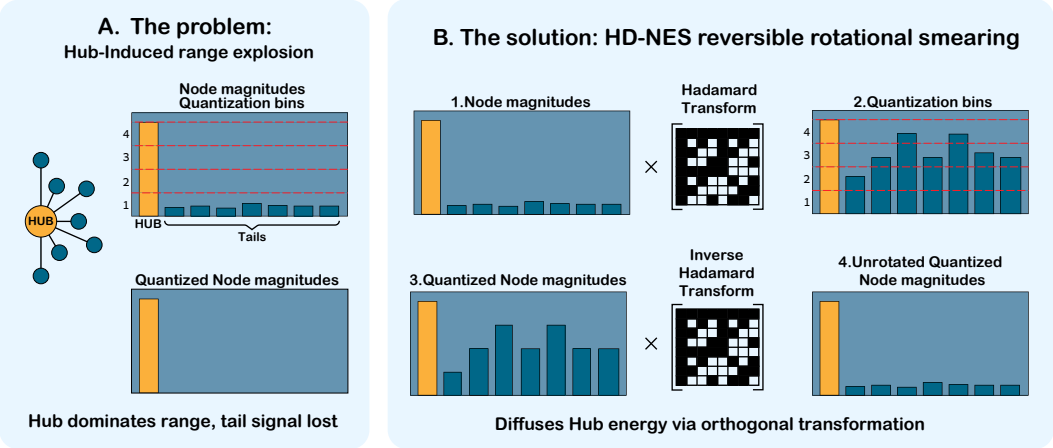


Figure 1: High-Degree Node Energy Smearing (*HD-NES*). (A) The Problem: Hub-induced range explosion occurs when hub nodes accumulate massive feature magnitudes that dominate the dynamic range, causing the fine-grained signal of tail nodes to be lost during quantization. (B) The Solution: *HD-NES* employs reversible rotational smearing via a Hadamard transform to diffuse the concentrated energy of hub nodes via orthogonal transformation. This topology-aware method applies lightweight, reversible rotation transformations to reduce activation variance while preserving information via reversibility, effectively mitigating hub-induced outliers and enabling accurate low-bit quantization.

ABSTRACT

Graph Neural Networks (GNNs) achieve state-of-the-art accuracy on irregularly structured data, yet efficient deployment on resource-constrained platforms remains challenging. This challenge is growing in importance as GNNs are increasingly incorporated into real-world products. In recent years, graphs underlying real-world applications have increased dramatically in size, making it necessary to operate under strict energy, latency, and memory constraints. Quantization has emerged as a promising approach to address these challenges. However, its accuracy degradation in message-passing GNNs is often severe. We show that this degradation is fundamentally topological: activation energy grows with node degree, causing hub nodes to dominate the dynamic range and reducing quantization resolution. Existing methods treat quantization error as a calibration or smoothness problem and fail to address this deterministic range explosion. We propose **High-Degree Node Energy Smearing (*HD-NES*)**, a topology-aware method that applies lightweight, reversible rotation transformations to diffuse the feature energy of hub nodes. By reducing activation variance while preserving information via reversibility, *HD-NES* mitigates hub-induced outliers and enables accurate low-bit quantization. *HD-NES* integrates into existing GNN quantization pipelines without architectural changes and consistently outperforms prior approaches. Code is available in the supplementary material.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become a foundational paradigm for learning over irregularly structured data and now underpin a broad range of high-impact applications. Initially demonstrating

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

strong performance on social Fan et al. (2019) and citation networks Kipf (2016); Veličković et al. (2018); Hamilton et al. (2017), GNNs have since been adopted in industrial-scale recommender systems Ying et al. (2018); Wang et al. (2019); He et al. (2020); Zhu et al. (2019), modeling for navigation and ETA prediction Derrow-Pinion et al. (2021), and scientific discovery workflows including computational chemistry and drug discovery Jumper et al. (2021); Stokes et al. (2020).

A defining characteristic of GNNs is their distinct memory profile: while parameter sharing yields compact model sizes, the message-passing mechanism requires substantial memory bandwidth to aggregate and store node features, particularly for large-scale graphs Hamilton et al. (2017); Chiang et al. (2019); Zeng et al. (2019).

Recent growth in graph size has made memory bandwidth a primary bottleneck for scalable GNN inference. This challenge is particularly timely given renewed interest in GNNs driven by the emergence of Graph Foundation Models (GFMs) Xia & Huang (2024); Finkelshtein et al. (2025). Unlike many architectures where inference cost correlates primarily with model size, GNN inference is fundamentally shaped by the input graph itself, since message passing must propagate information over potentially massive edge sets. As a result, even models with very few parameters can incur extremely large inference cost on real-world graphs. For example, a standard two-layer Graph Convolutional Network (GCN) on the large-scale `ogbn-papers` dataset requires 3.6 TeraOPs for a single inference pass despite having only $\approx$19k parameters. This already exceeds the forward-pass compute of ViT-Giant Zhai et al. (2022), a $\sim$1B-parameter model. A broader comparison to Transformer baselines is provided in Appendix Figure 8.

This observation highlights a critical bottleneck: although GNN architectures are parameter-efficient, their runtime is mostly affected by the graph size as large graphs structures can make high-precision inference computationally prohibitive. To mitigate these costs, model quantization, a map from high-precision floating-point values to lower-bit representations (e.g., 8-bit or 4-bit), has emerged as an effective optimization strategy. By compressing weights and activations, quantization reduces memory footprint and bandwidth demands while leveraging the high-throughput integer arithmetic capabilities of modern hardware.

However, the naïve application of uniform quantization often leads to severe accuracy degradation in GNNs due to the *outlier phenomenon*Tailor et al. (2021). Real-world graphs typically follow a power-law distribution, where a minority of "hub" nodes possess extremely high degrees. During message passing, these hubs accumulate feature with magnitudes far exceeding those of the average node, stretching the quantization range, effectively masking fine-grained information with quantization noise.

While recent works have attempted to mitigate this via mixed-precision or smoothness regularization Tailor et al. (2021); Moustafa et al. (2025); Zhu et al. (2023); Wang et al. (2023), they do not directly address the underlying highly skewed distribution resulting from graph topology.

We introduce **H**igh **D**egree **N**ode **E**nergy **S**mearing (*HD-NES*). *HD-NES* utilizes a topology-aware rotational transformation to redistribute the massive energy of hub nodes Figure 1. This approach homogenizes the activation distribution, rendering it amenable to high-precision quantization.

Our contributions are as follows:

- We introduce *HD-NES*: a quantization method that utilizes randomized, reversible rotations to diffuse hub energy across different nodes. This effectively smooths the heavy-tailed activation distribution of power-law graphs, enabling robust quantization.
- We propose a variant of *HD-NES* for edge smearing. We justify this design by showing that, specifically in Graph Attention Networks (GATs), the underlying limitation stems from attention mechanics rather than node energy.
- We validate the proposed method on various graph benchmarks, consistently improving quantization performance.
- We provide a theoretical complexity analysis validating our approach using an optimized kernel implementation. We demonstrate that *HD-NES* introduces negligible runtime overhead compared to standard GNN computation, making it practical for real-world deployment.

2 BACKGROUND

2.1 QUANTIZATION AWARE TRAINING (QAT)

Model quantization reduces the computational complexity and memory footprint of deep neural networks by mapping high-precision 32-bit floating-point (FP32) tensors to lower-bit representations (e.g., INT8). While post-training quantization (PTQ) performs this mapping on a pre-trained model, Quantization Aware Training (QAT) simulates quantization noise during the training process. This allows the model to adapt its parameters to minimize the loss induced by the lower precision, typically yielding higher accuracy for low-bit settings.

Uniform Quantization. A common quantization scheme is uniform symmetric quantization, which maps a real-valued input x to an integer value q within a discrete set. For a bit-width b, the quantization grid is defined over the integer range $[-2^{b-1}, 2^{b-1} - 1]$. The simulated quantized value (or de-quantized approximation), denoted as $\hat{x}$, is computed as $\hat{x} = Q(x) = S \cdot \text{clip}\left(\left\lfloor \frac{x}{S} \right\rceil, -2^{b-1}, 2^{b-1} - 1\right)$, where S is the scaling factor (step size) derived from the dynamic range such that $S \approx \frac{\max(|x|)}{2^{b-1}-1}$, and $\lfloor \cdot \rceil$ denotes the rounding-to-nearest operation, defined as $\lfloor z \rceil = \lfloor z + \frac{1}{2} \rfloor$, and $\text{clip}(z, n, m) = \min(\max(z, n), m)$.

2.2 ORTHOGONAL MATRICES AND THE HADAMARD TRANSFORM

We utilize orthogonal matrices, $\mathbf{R}$, defined by the property $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$. Geometrically, applying $\mathbf{R}$ corresponds to a rotation of the coordinate system that preserves Euclidean norms. A fundamental example of such a basis change is the Discrete Fourier Transform (DFT). The DFT can be viewed as a rotation into the frequency domain, which effectively "delocalizes" or spreads energy: a sparse signal (like a single spike) in the input domain becomes uniform in the frequency domain.

The **Walsh-Hadamard transform** acts as a real-valued, binary analog to the Fourier transform. A Hadamard matrix $\mathbf{H}$ is an orthogonal matrix with entries in $\{+1, -1\}$.

Similar to the Fast Fourier Transform (FFT), the structured nature of $\mathbf{H}$ allows for matrix-vector multiplication in $\mathcal{O}(d \log d)$ operations. This makes it a highly efficient method for rotating data to reduce peak activation values (outliers) without the computational cost of general matrix multiplication.

3 GNN QUANTIZATION ANALYSIS

The fundamental challenge in quantizing Graph Neural Networks (GNNs) arises from the intrinsic coupling between graph topology and feature activation magnitudes. Unlike Computer Vision (CV) or Natural Language Processing (NLP), where activation distributions are primarily governed by learned weights and input content rather than explicit sample-dependent structure He et al. (2016); Vaswani et al. (2017), GNN activation statistics are heavily influenced by the underlying graph connectivity Kipf (2016); Xu et al. (2019). This structural dependency creates a severe divergence in quantization requirements for high-degree "hub" nodes versus low-degree "tail" nodes.

3.1 THE HUB NODE DILEMMA: RANGE EXPLOSION

In Message Passing GNNs, the aggregation mechanism accumulates features from a node's local neighborhood. For architectures employing sum-aggregation, such as GIN (Xu et al., 2019) or general unnormalized MPNNs (Gilmer et al., 2017), the magnitude of the aggregated feature vector h_v^{agg} correlates directly with the node degree d_v, following $\mathbb{E}[\|h_v^{\text{agg}}\|] \propto d_v \cdot \mathbb{E}[\|h_u\|]$. Real-world graphs, such as citation networks or social graphs, typically exhibit power-law degree distributions $P(d) \propto d^{-\gamma}$ (Empirical analysis at Section E). Unlike exponential distributions, power laws are heavy-tailed, implying a non-negligible probability of nodes possessing degrees orders of magnitude larger than the average. This results in the presence of extreme "hub" nodes.

During inference, these hubs generate extreme activation outliers. In standard uniform symmetric quantization, the scaling factor S (step size) is determined by the global dynamic range to prevent clipping as $S = \frac{\max_{v \in \mathcal{V}} |h_v|}{2^{b-1}-1}$. Since $\max |h_v|$ is dictated by a minority of extreme hub nodes,

S becomes excessively large. While this wide range accommodates the hubs, it results in an overly coarse quantization grid for the vast majority of the graph nodes, effectively squashing their representations to zero.

3.2 The Tail Node Problem: Signal Collapse

The vast majority of nodes in power-law graphs are low-degree "tail" nodes. Consequently, their aggregated feature magnitudes are significantly smaller than those of hub nodes. The primary quantization challenge for tail nodes is not range, but resolution.

When the quantization scale S is inflated by hub-driven outliers, fine-grained feature variations in tail nodes often fall below the resolution of the smallest quantization interval. For a tail node feature x_{tail}, if the signal strength is weak relative to the scale S, such that $|x_{\text{tail}}| < \frac{S}{2} \implies Q(x_{\text{tail}}) = S \cdot \left\lfloor \frac{x_{\text{tail}}}{S} \right\rceil = 0$, we term this phenomenon *Quantization Bin Collapse*. This effect erases the semantic information for the majority of nodes in the graph. Standard uniform quantization thus creates a paradox: the model retains precision for structural outliers (hubs) while failing to resolve the features of the widespread tail nodes, leading to severe performance degradation.

3.3 The Expressivity–Quantization Trade-off

A natural question is whether the hub-induced range explosion can be avoided by modifying the aggregation rule itself, for example through degree normalization or mean aggregation. While such choices reduce activation magnitude, they do not eliminate the underlying problem and, in some cases, sacrifice expressive power.

Injective aggregation is necessary for a GNN to match the discriminative power of the Weisfeiler-Lehman (WL) test Xu et al. (2019). Sum aggregation satisfies this requirement by preserving multiset cardinality, but in doing so induces activation magnitudes that scale linearly with node degree. As shown in Section 3.1, this directly leads to extreme outliers that dominate the quantization range.

Degree-normalized architectures such as GCNs Kipf (2016) partially alleviate this effect. By scaling messages with inverse degree factors, GCNs reduce activation growth from linear to sub-linear in the node degree (formalized in Section 3.4). However, this mitigation is insufficient in practice: on power-law graphs, even sub-linear growth yields large dynamic range gaps between hub and tail nodes, which remain detrimental under low-bit quantization. Thus, normalization softens but does not resolve the topology-induced quantization bottleneck.

More aggressive normalization or mean aggregation can fully bound activation magnitudes, but at the cost of collapsing degree information and weakening structural discrimination. This reveals a fundamental trade-off: preserving expressivity induces degree-dependent energy concentration, while suppressing energy sacrifices information.

3.4 Scaling Analysis

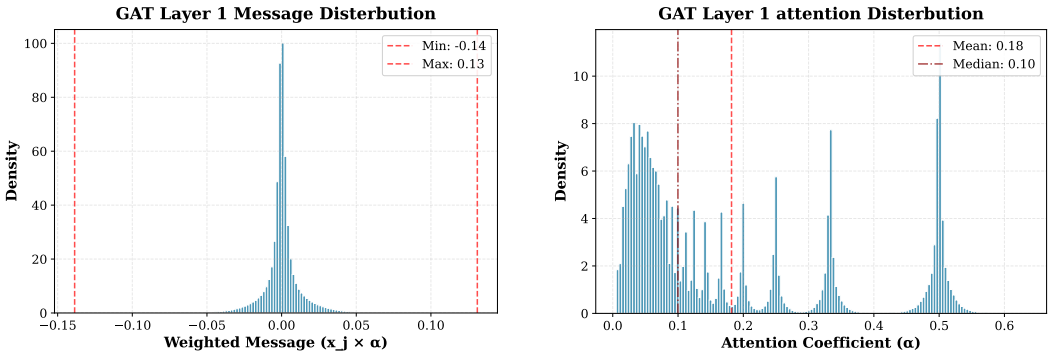
We formalize how different aggregation strategies affect the scaling of the activations Y_n with respect to node degree n. Let incoming messages X_i be random variables with finite variance.

3.4.1 GIN: Linear Scaling

GIN employs sum-aggregation, $Y_n^{\text{GIN}} = \sum_{i=1}^{n} X_i$. Assuming weakly correlated inputs, the variance of the aggregation scales linearly with degree n as $\text{Var}(Y_n^{\text{GIN}}) \sim \mathcal{O}(n)$. This unbounded linear growth ensures that as graph density or node degree increases, the dynamic range of activations expands indefinitely, predisposing high-degree nodes to become extreme outliers.

3.4.2 GCN: Sublinear Scaling

GCN incorporates a symmetric normalization factor, acting as a structural scale control. The update rule is $h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{1}{\sqrt{d_v d_u}} W^{(l)} h_u^{(l)}\right)$, where $h_v^{(l+1)}$ is the feature vector of node v at layer $l+1$, σ denotes a non-linear activation function, and $\mathcal{N}(v) \cup \{v\}$ represents the set of neighbors



(a) Layer 1 Weighted messages

(b) Layer 1 Attention coefficients

Figure 2: Layer 1 distributions in GAT on Pubmed. Weighted edge messages are leptokurtic, while attention coefficients are sparse and multimodal, indicating that GAT's quantization bottleneck stems from edge-level sparsity rather than degree-induced range explosion.

of v including itself. The terms d_v and d_u refer to the degrees of nodes v and u, respectively, while $W^{(l)}$ is the learnable weight matrix and $h_u^{(l)}$ is the input feature vector from neighbor u.

For a high-degree node v, incoming messages are down-weighted by $1/\sqrt{d_v}$. Assuming $\|W^{(l)}h_u^{(l)}\| \leq M$ and $d_u \geq 1$, the aggregation magnitude is deterministically bounded as $\left\|\sum_{u\in\mathcal{N}(v)} \frac{1}{\sqrt{d_v}} W^{(l)}h_u^{(l)}\right\| \leq \sqrt{d_v}M$. This demonstrates that GCN aggregation scales sub-linearly, $\mathcal{O}(\sqrt{n})$, significantly reducing the outlier gap between hub and tail nodes compared to GIN and other architectures relying on summation aggregation (e.g., MPNNGilmer et al. (2017)).

3.4.3 GAT: Constant Scaling

Attention-based mechanisms (GAT) perform a weighted mean where the coefficients α sum is one: $Y_n^{\text{ATTN}} = \sum_{i=1}^n \alpha_i X_i$, s.t. $\sum_{i=1}^n \alpha_i = 1$. Mathematically, this yields constant variance scaling $\mathcal{O}(1)$, effectively strictly bounding the output magnitude independent of node degree n. While this successfully prevents the "Range Explosion" seen in GIN, attention introduces a different challenge: *attention sparsity*. Empirically (Figure 2 and Section D.1), attention coefficients cluster at discrete values rather than forming a smooth distribution, as the model attends selectively to few edges. This sparsity propagates through message weighting ($\alpha \cdot x_j$), producing leptokurtic distributions with extreme concentration near zero and rare outliers (Figure 2 and Section D.2). Uniform quantization wastes precision on these sparse tail regions.

4 Methodology: *HD-NES*

We propose *HD-NES*, a quantization strategy that decouples activation magnitude from node degree via reversible energy redistribution. We apply the Hadamard transform introduced in Section 2.2 to redistribute activation energy across the node axis. The key insight is that by rotating the node-level feature matrix into a frequency-like domain before quantization, we can "smear" the concentrated energy of hub nodes across all nodes, thereby flattening the dynamic range and enabling finer quantization resolution.

Notation. Let $\mathbf{X} \in \mathbb{R}^{N\times D}$ denote node feature matrix, where N is the number of nodes and D is the feature dimension. Let B denote the block size for the Hadamard transform, which must be a power of 2.

4.1 Rotational Energy Redistribution

We define the *HD-NES* operation as a sequence of centering, block-wise randomized Hadamard rotation, and quantization.

Transform Operator. We construct an orthogonal transformation Ψ that operates on blocks of size B. First, features are zero-centered via $\tilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}$ to maximize outlier delocalization. To enable block-wise processing, $\tilde{\mathbf{X}}$ is zero-padded to ensure the first dimension is divisible by B. We then slice the matrix along this dimension to form K contiguous blocks $\mathbf{X}^{(k)} \in \mathbb{R}^{B \times D}$. We apply a Randomized Hadamard transform to each block: $\mathbf{Z}^{(k)} = \Psi(\mathbf{X}^{(k)}) = \mathbf{H}_B \mathbf{D} \mathbf{X}^{(k)}$ where $\mathbf{H}_B$ is the normalized Walsh-Hadamard matrix and $\mathbf{D} \in \{-1, +1\}^{B \times B}$ is a diagonal Rademacher matrix. This rotation "smears" the energy of any single hub node uniformly across the B dimensions of the subspace, reducing the peak magnitude. The centering operation is analogous to removing the DC component in signal processing, ensuring that the transform focuses on distributional variance rather than offset.

Quantization and Inverse. Quantization is performed in the rotated basis $\mathbf{Z}$, where the dynamic range is compressed. The reconstructed approximation $\hat{\mathbf{X}}$ is obtained by applying the inverse rotation and adding back the subtracted mean $\boldsymbol{\mu}$: $\hat{\mathbf{Z}} = Q(\mathbf{Z}), \quad \hat{\mathbf{X}}^{(k)} = \mathbf{D}\mathbf{H}_B \hat{\mathbf{Z}}^{(k)} + \boldsymbol{\mu}$ Crucially, since Ψ is orthogonal, the quantization error introduced in $\mathbf{Z}$ is distributed uniformly across all nodes in the block upon inversion, rather than concentrating on the hubs.

Node-Level Smearing (GIN/GCN). For GNNs where aggregation scales with degree, *HD-NES* is applied on the nodes axis, smearing energy across nodes.

Edge-Level Smearing (GAT). While node-level smearing addresses degree-dependent energy growth, GATs require a distinct approach due to their constant aggregation scaling $\mathcal{O}(1)$. As identified in Section 3.4, the bottleneck is the structural sparsity of the weighted messages $\mathbf{m}_{ij} = \alpha_{ij} \mathbf{W}\mathbf{h}_j$. The attention coefficients α_{ij} exhibit a multimodal distribution, rendering the message tensor $\mathbf{M} \in \mathbb{R}^{|\mathcal{E}| \times D}$ highly leptokurtic. We therefore apply *HD-NES* along the *edge dimension* $|\mathcal{E}|$. By applying the rotation Ψ to blocks of edges, we mix high-energy messages (large α_{ij}) with low-energy background edges. This diffuses attention spikes, transforming the leptokurtic distribution into a smoother, Gaussian-like form that maximizes quantization grid utilization.

Complexity. *HD-NES* introduces negligible computational overhead by leveraging the Fast Walsh-Hadamard Transform (FWHT). The transformation scales as $\mathcal{O}(ND \log B)$ and relies solely on addition and subtraction operations, avoiding expensive floating-point multiplications. We provide a detailed theoretical derivation and real-hardware runtime benchmarks in Section A.

4.2 Theoretical Analysis

We provide a theoretical bound on how *HD-NES* mitigates quantization error by compressing the dynamic range. We defer the full proof of Lemma 4.1 to Appendix I, and the full proof of Theorem 4.2 is given in Appendix H.

Peak Reduction via Rotation. We employ the Randomized Hadamard Transform, $\Psi(\mathbf{x}) = \mathbf{H}_B \mathbf{D} \mathbf{x}$. Since $\mathbf{H}_B$ is orthogonal, it preserves the ℓ_2 norm (energy) of the signal. However, it alters the ℓ_∞ norm. By spreading the energy of high-magnitude components across the block, we obtain the following bound:

Lemma 4.1 (Randomized Hadamard Peak Bound). *Let $\mathbf{x} \in \mathbb{R}^B$ be any fixed vector. Let $\mathbf{D}$ be a random diagonal matrix with i.i.d. Rademacher entries, independent of $\mathbf{x}$. Then, with probability at least $1-\delta$ over $\mathbf{D}$, the rotated vector $\mathbf{z} = \mathbf{H}_B \mathbf{D} \mathbf{x}$ satisfies the inequality $\|\mathbf{z}\|_\infty \leq \sqrt{\frac{2 \log(2B/\delta)}{B}} \|\mathbf{x}\|_2$.*

The proof of this bound is well-established in randomized linear algebra Ailon & Chazelle (2006); Tropp (2011). Crucially, the randomness in Lemma 4.1 arises solely from the Rademacher rotation and does not assume independence of node features, which may be correlated due to message passing.

Quantization Error Minimization. Uniform quantization creates an error determined by the step size S, which must accommodate the entire dynamic range.

Theorem 4.2 (MSE Reduction). *Let the step size be $S = 2\|\mathbf{z}\|_\infty / (2^b - 1)$. Since Ψ is orthogonal, it preserves the Euclidean norm of the quantization noise. The expected reconstruction error is $\mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2] = \mathbb{E}[\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2] \approx \frac{BS^2}{12} \propto \|\mathbf{z}\|_\infty^2$.*

Table 1: Comparison Between *HD-NES* and QAT with uniform quantization (5 runs ± Std. Dev.).

Quant Scheme	Arch.	Node Classification (Acc %)				Link Prediction (Mrr %)
		Cora ↑	Citeseer ↑	Pubmed ↑	OGB-arxiv ↑	Ogbl-Citation2
FP32	GCN	81.5 ± 0.7	71.4 ± 0.7	78.2 ± 0.7	71.6 ± 0.1	68.0 ± 0.3
	GIN	80.2 ± 0.5	64.2 ± 0.7	76.4 ± 0.7	68.6 ± 0.4	78.9 ± 0.5
	GAT	83.1 ± 0.9	70.1 ± 0.8	77.7 ± 0.5	71.5 ± 0.3	77.4 ± 0.4
	GATv2	83.9 ± 0.8	71.4 ± 0.6	78.2 ± 0.6	71.8 ± 0.2	81.4 ± 0.7
QAT (W8A8)	GCN	78.3 ± 0.9	67.4 ± 0.8	75.2 ± 0.8	71.1 ± 0.2	40.4 ± 0.4
	GIN	79.6 ± 1.1	62.3 ± 0.9	60.5 ± 0.6	63.5 ± 0.3	38.6 ± 0.2
	GAT	78.6 ± 0.8	69.1 ± 0.8	77.0 ± 0.7	55.8 ± 0.4	22.7 ± 0.5
	GATv2	80.6 ± 0.8	71.3 ± 0.7	77.8 ± 0.8	67.6 ± 0.4	45.2 ± 0.7
QAT (W4A4)	GCN	73.4 ± 1.0	63.6 ± 1.1	74.2 ± 0.8	31.4 ± 0.2	36.8 ± 0.3
	GIN	58.6 ± 1.2	56.2 ± 1.0	60.5 ± 1.1	31.3 ± 0.1	1.9 ± 0.1
	GAT	49.2 ± 0.9	46.3 ± 0.9	55.1 ± 0.9	21.8 ± 0.2	0.3 ± 0.04
	GATv2	61.7 ± 1.1	37.0 ± 1.3	53.8 ± 1.1	21.1 ± 0.2	1.7 ± 0.07
HD-NES (W8A8)	GCN	79.1 ± 0.6 (+0.8%)	69.4 ± 0.7 (+2.0%)	76.4 ± 0.8 (+1.2%)	71.2 ± 0.1 (+0.1%)	65.6 ± 0.4 (+25.2%)
	GIN	80.2 ± 0.5 (+0.6%)	64.2 ± 0.8 (+1.9%)	74.4 ± 0.7 (+13.9%)	69.1 ± 0.3 (+5.6%)	59.1 ± 0.3 (+20.5%)
	GAT	80.7 ± 0.4 (+2.1%)	69.7 ± 0.7 (+0.6%)	77.9 ± 0.6 (+0.9%)	67.8 ± 0.4 (+12%)	77.1 ± 0.5 (+54.4%)
	GATv2	82.4 ± 0.6 (+1.8%)	71.3 ± 0.6 (+0.0%)	77.9 ± 0.7 (+0.1%)	70.6 ± 0.3 (+3.0%)	77.6 ± 0.5 (+32.4%)
HD-NES (W4A4)	GCN	79.4 ± 0.7 (+6%)	69.5 ± 0.9 (+5.9%)	76.1 ± 0.9 (+1.9%)	37.8 ± 0.2 (+6.4%)	57.2 ± 0.3 (+20.4%)
	GIN	67.2 ± 0.7 (+8.6%)	60.4 ± 0.8 (+4.2%)	74.4 ± 0.9 (+13.9%)	40.7 ± 0.2 (+9.4%)	5.2 ± 0.2 (+3.3%)
	GAT	73.9 ± 0.6 (+24.7%)	68.5 ± 0.9 (+22.2%)	76.1 ± 0.8 (+21%)	35.1 ± 0.5 (+13.3%)	41.2 ± 0.4 (+40.9%)
	GATv2	80.5 ± 0.5 (+18.8%)	67.9 ± 1.1 (+30.9%)	73.5 ± 0.7 (+19.7%)	56.1 ± 0.6 (+35.0%)	45.7 ± 0.8 (+44.0%)

Table 2: GIN quantization results: accuracy (%) ± standard deviation. Parentheses show gains over the 4-bit DQ baseline Tailor et al. (2021), with both weights and activations quantized.

Quant.Scheme	Cora	Reddit Binary
FP32	77.6 ± 1.1	92.0 ± 1.5
DQ	69.9 ± 3.4	81.3 ± 4.4
HD-NES + DQ	76.1 ± 1.4 (**+7.2**)	85.9 ± 0.2 (**+4.6**)

Table 3: GCN quantization results: accuracy (%) ± standard deviation. Parentheses show gains over the 2-bit SMP baseline Wang et al. (2023), with both weights and activations quantized.

Quant. Scheme	Cora	Reddit Binary
FP32	80.79 ± 1.54	94.04 ± 0.18
SMP	74.56 ± 2.25	85.19 ± 1.36
HD-NES + SMP	76.46 ± 1.4 (**+1.9**)	86.96 ± 0.2 (**+1.77**)

Combining this with Lemma 4.1, reducing the peak magnitude $\|\mathbf{z}\|_\infty$ via rotation directly minimizes the global quantization noise power.

5 Experiments

We evaluate the efficacy of *HD-NES* across multiple graph benchmarks and architectures. Our experiments aim to address several key research questions. First, **(RQ1)** investigates whether *HD-NES* enables robust low-bit quantization (e.g., W4A4) compared to standard quantization baselines. Second, **(RQ2)** explores if *HD-NES* can be integrated into existing state-of-the-art quantization frameworks to provide additive gains. Third, **(RQ3)** examines whether the empirical results align with our theoretical scaling analysis regarding node degree and aggregation types, such as GIN vs. GCN. Finally, **(RQ4)** assesses the contribution of specific design choices, including the rotation axis and block size.

5.1 Experimental Setup

Implementation Details. We implement all models using the PyTorch framework and the PyTorch Geometric (PyG) library. The *HD-NES* module is integrated as a differentiable operator within standard GNN layers. Consistent with the methodology described in Appendix B, the insertion point varies by architecture: for GCN and GIN, the transformation is applied to the node feature matrix; for GAT, it is applied to the edge-level message tensor.

Hyperparameters and Optimization. Our evaluation controls for two primary hyperparameters introduced by the *HD-NES* mechanism. First, we consider the **Hadamard Block Size** (B), experimenting with block sizes $B \in \{2^k \mid k \in \{3, \dots, 10\}\}$ to effectively balance smearing granularity with computational overhead. Additionally, we account for **Layer Sensitivity**, as empirical analysis reveals that outlier sensitivity varies across network depth; consequently, we treat the application of *HD-NES* to specific layers as a structural hyperparameter. All quantization experiments simulate

low-precision inference using Quantization Aware Training (QAT) with Straight-Through Estimators (STE). Detailed hyperparameter configurations are provided in Appendix B.

5.2 Performance on Low-Bit Regimes (RQ1)

To answer **(RQ1)**, we compare *HD-NES* against standard uniform Quantization Aware Training (QAT) across W8A8 and W4A4 regimes. Table 1 presents the comparative performance of GCN, GIN, and GAT architectures.

We observe that while standard QAT (W8A8) maintains reasonable fidelity to FP32 baselines, performance degrades precipitously in the W4A4 regime due to the bin collapse phenomenon described in Section 3. *HD-NES* consistently mitigates this degradation. Notably, in the ultra-low bit regime (W4A4), *HD-NES* recovers significant accuracy on the `Ogbl-Citation2` dataset, improving GCN performance by **20.4%** and GAT performance by **40.9%** relative to standard QAT. Furthermore, our specialized Edge-Level Smearing for GAT effectively stabilizes the attention mechanism, addressing the resolution loss in multimodal distributions.

5.3 Integration with Existing Frameworks (RQ2)

To address **(RQ2)**, we validate whether *HD-NES* is complementary to existing specialized GNN quantization frameworks. We integrated our method into DegreeQuant (DQ) Tailor et al. (2021) and Smoothness-aware Message Passing (SMP) Wang et al. (2023).

Table 2 compares vanilla DQ against DQ augmented with *HD-NES* for a GIN architecture. Adding *HD-NES* provides a significant improvement, boosting accuracy on Cora by **7.2%** and on Reddit Binary by **4.6%**. Similarly, Table 3 demonstrates that integrating *HD-NES* with SMP on a GCN architecture improves Cora accuracy by **1.9%**. These results confirm that smearing energy is orthogonal to topology-aware calibration and smoothness regularization, providing additive gains when combined.

5.4 Consistency with Scaling Analysis (RQ3)

To address **(RQ3)**, we analyze the performance divergence between architectures employing different aggregation rules. As shown in Table 1 (W4A4 regime), GIN collapses to near random performance (1.9% MRR) on `Ogbl-Citation2` under standard quantization. This observation corroborates our analysis in 3.4, confirming that linear $\mathcal{O}(n)$ variance scaling results in severe range explosion. Specifically on `Ogbl-Citation2`, the degree distribution is sufficiently skewed that the resulting dynamic range saturates the limited representational capacity of INT4; notably, this variance is so extreme that signal recovery remains challenging even when applying *HD-NES*.

In contrast, GCN retains 36.8% accuracy in the same setting, supporting the hypothesis that $\mathcal{O}(\sqrt{n})$ normalization provides intrinsic, albeit partial, immunity to degree-induced range expansion. Nevertheless, *HD-NES* provides significant additive gains over this baseline, demonstrating that even with sub-linear scaling, topological outliers constitute a non-trivial bottleneck that energy smearing effectively mitigates.

5.5 Ablation Study (RQ4)

Finally, to answer **(RQ4)**, we conduct ablation studies to isolate the contribution of the specific structural design choices of *HD-NES*.

Sensitivity to Block Size (B). We evaluate the performance of *HD-NES* on GIN (OGB-Arxiv) across varying block sizes. As detailed in Appendix C, performance peaks at $B = 64$. This indicates that the receptive field of the rotation must be large enough to absorb outliers, but local enough to preserve graph structure. Increasing B further yields diminishing returns.

The Necessity of Node-Axis Rotation. We hypothesize that GNN quantization bottlenecks are topological (variance between nodes) rather than semantic (variance between features). We tested a variant of *HD-NES* that applies the Hadamard transform across feature channels rather than the node axis. This variant resulted in severe accuracy degradation (approx. -10% drop), as detailed in

Appendix C. This confirms that mixing features destroys semantic alignment without resolving the degree-induced magnitude disparity, validating our choice of node-axis rotation.

Aggregation Expressivity. We demonstrate in Appendix F that sum aggregation (GIN) maintains a representational advantage over mean aggregation under 4-bit constraints, and that *HD-NES* can be used to mitigate the resulting range explosion.

Generalization Across Graph Types. We evaluate *HD-NES* on diverse graph topologies, including heterophilic graphs such as Chameleon and Squirrel. Results in Appendix G show consistent mitigation of quantization degradation across structural regimes, while Section F discusses its effectiveness in inductive settings.

6 Related Work

Quantization for Graph Neural Networks. Recent GNN quantization work largely targets *how* to allocate precision or calibrate scales, rather than the topology-driven mechanisms analyzed in Section 3. DegreeQuant (DQ) Tailor et al. (2021) improves GNN quantization by stochastically protecting high-degree nodes from quantization during training and using percentile-based range tracking, but it does not alter the aggregation-induced magnitude growth that causes hub-dominated ranges and tail-node resolution loss at inference (Section 3). More recent works have focused on the interplay between graph smoothness and quantization error. Smoothness-aware Message Passing (SMP) Wang et al. (2023) proposes a regularization technique to enforce smoothness in the latent space, arguing that smoother signals are more robust to quantization noise. While SMP mitigates noise sensitivity, it does not directly address the *magnitude* outliers generated by injective aggregation functions (like sum-pooling in GINs), furthermore, this method requires large amount of layers for the smoothness process. Other approaches, such as MixQ Moustafa et al. (2025) and A2Q Zhu et al. (2023), employ mixed-precision strategies, assigning higher bit-widths to sensitive nodes or layers. While this preserves accuracy, it requires specialized hardware support for mixed-precision arithmetic and complicates the deployment of homogeneous integer-only kernels. In contrast, our approach targets a uniform low-bit representation by reshaping the underlying distribution rather than varying the precision.

Outlier Mitigation and Rotational Invariance. The issue of activation outliers has been extensively studied in the context of Large Language Models (LLMs). Recent methods like QuaRot Ashkboos et al. (2024) and Hadanorm Federici et al. (2025) demonstrate that applying randomized orthogonal rotations (e.g., Hadamard or Walsh-Hadamard transforms) to weight and activation matrices can suppress outliers by delocalizing signal energy. However, these methods typically operate on independent token or channel dimensions assuming i.i.d. distributions. They do not account for the structural dependencies inherent in message passing, where outliers are not merely statistical anomalies but deterministic consequences of the graph topology (hub nodes). Our work extends the principle of rotational energy smearing to the graph domain. To the best of our knowledge, *HD-NES* is the first method to apply topology-aware, block-wise rotations to decouple node degree from activation magnitude, enabling high-fidelity quantization for both sum-aggregation architectures and attention-based mechanisms.

7 Discussion

We have demonstrated that quantization bottlenecks in GNNs are fundamentally topological. By decoupling activation magnitude from node degree via *HD-NES*, we resolve the trade-off between model expressivity and numerical precision. Our results highlight two critical findings. First, for sum-aggregation architectures like GIN, *HD-NES* prevents the range explosion that typically creates a choice between high expressivity and quantizability. Second, and perhaps more surprisingly, we show that even structurally normalized architectures like GCN suffer from topological outliers in ultra-low bit regimes (W4A4). While GCN's inverse-degree scaling dampens outliers, it does not eliminate them; *HD-NES* provides the necessary smoothing to recover accuracy where standard normalization falls short. Finally, our GAT extension shows that rotational redistribution remains valuable where analytical normalization is unavailable. Its block size B remains a tunable hyperparameter (Appendix A); future work could select B adaptively from local degree centrality to balance hub-energy smearing against feature homogenization.

AI USE STATEMENT

In this work, we used generative AI tools for / to:

- Assistance in the writing of proofs.
- Implement methods

We have not used generative AI tools for / to:

- Help develop theoretical models or conceptual frameworks.
- Formulate mathematical claims.
- Provide critical ingredients for proving mathematical claims.
- Propose or refine hypotheses.
- Design or provide feedback on research methodology or experiments.
- Assist with translation.
- Clean and reformat dataset.
- Interpret results.
- Generation of synthetic data sets.
- Support qualitative and thematic data analysis.

Additionally, we used generative AI tools for / to:

- Create or edit software code.
- Summarize or analyse existing literature, and identify relevant literature.
- Draft parts of a research paper.
- Edit a research paper to improve readability.

We have reviewed all AI-assisted work. LLM-generated code was verified and tested rigorously for validity and correctness. All written LLM-generated text appearing in the submitted manuscript was reviewed and approved by the authors. All suggested literature was reviewed and cross-referenced by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the source code and computational kernels used in this work as supplementary material. These materials include the implementation of our method and the kernels used for its execution, enabling independent reproduction and verification of our work.

REFERENCES

Hadacore: Tensor core accelerated hadamard transform kernel. `https://pytorch.org/blog/hadacore/`. Software kernel / project page.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 557–563, 2006.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Roberto L. Castro, Saleh Ashkboos, and Soroush Tabesh. Int8 gemm with pytorch interface, 2024. URL `https://github.com/IST-DASLab/gemm-int8`.

Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 257–266, 2019.

Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3767–3776, 2021.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pp. 417–426, 2019.

Marco Federici, Riccardo Del Chiaro, Boris van Breugel, Paul Whatmough, and Markus Nagel. Hadanorm: Diffusion transformer quantization through mean-centered transformations. *arXiv preprint arXiv:2506.09932*, 2025.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Ben Finkelshtein, İsmail İlkan Ceylan, Michael Bronstein, and Ron Levie. Equivariance everywhere all at once: A recipe for graph foundation models, 2025.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

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

Samir Moustafa, Nils Kriege, and Wilfried N Gansterer. Efficient mixed precision quantization in graph neural networks. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 4038–4052. IEEE, 2025.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2023.

Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

Shyam A. Tailor, Javier Fernandez-Marques, and Nicholas D. Lane. Degree-quant: Quantization-aware training for graph neural networks. In *International Conference on Learning Representations*, 2021.

Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

Shuang Wang, Bahaeddin Eravci, Rustam Guliyev, and Hakan Ferhatosmanoglu. Low-bit quantization for deep graph neural networks with smoothness-aware message propagation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 2626–2636, 2023.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 165–174, 2019.

Lianghao Xia and Chao Huang. Anygraph: Graph foundation model in the wild. *arXiv preprint arXiv:2408.10700*, 2024.

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

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graph-saint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113, 2022.

Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. Aligraph: A comprehensive graph neural network platform. *arXiv preprint arXiv:1902.08730*, 2019.

Zeyu Zhu, Fanrong Li, Zitao Mo, Qinghao Hu, Gang Li, Zejian Liu, Xiaoyao Liang, and Jian Cheng. A2q: Aggregation-aware quantization for graph neural networks. *arXiv preprint arXiv:2302.00193*, 2023.

A Computational Efficiency and Runtime Analysis

A.1 Theoretical Complexity

The computational cost of Graph Neural Networks is typically dominated by the message-passing phase, which scales with the number of edges, $\mathcal{O}(|\mathcal{E}|D)$. In contrast, ***HD-NES*** operates on the node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ by applying a randomized Hadamard transform along the node dimension (a left-multiplication).

We utilize a block-wise approach, partitioning the N nodes into $\lceil N/B \rceil$ blocks of size B. The Fast Walsh–Hadamard Transform (FWHT) processes a vector of size B in $\mathcal{O}(B \log B)$ operations. Since the transform is applied independently to each of the D feature columns within a block, the cost per block is $D \cdot \mathcal{O}(B \log B)$.

The total complexity for applying ***HD-NES*** to the entire graph is the product of the number of blocks and the cost per block:

$$\mathcal{O}\left(\underbrace{\frac{N}{B}}_{\text{Blocks}} \cdot \underbrace{D \cdot (B \log B)}_{\text{Cost per block}} \right) = \mathcal{O}(N D \log B). \tag{1}$$

Given that $\log B$ is a small constant (e.g., $\log 64 = 6$) and generally $|\mathcal{E}| \gg N$, the overhead introduced by ***HD-NES*** is negligible compared to the quadratic costs of dense layers or the irregular memory access patterns of message aggregation.

A.2 Implementation

The Hadamard transform is particularly amenable to hardware acceleration as it requires only additions and subtractions, eliminating the need for expensive floating-point multiplications. On modern GPUs, the operation is memory-bandwidth bound rather than compute-bound, allowing the transform to often be fused with the memory load operations preceding quantization.

For our implementation, we adapted the `hadacore`[1] library. While the original implementation supports fp16 right-sided Hadamard transforms (mixing features), we modified the kernel to perform an 8-bit block-wise *left* Hadamard transform.

Moreover, the Hadamard transform is widely used in image compression and video encoding; consequently, there is significant research on how to deploy it optimally on different hardware architectures.

A.3 Empirical Benchmarks

To assess the practical overhead of ***HD-NES***, we benchmarked the inference latency of a single GCN layer with hidden size of 16 on the Reddit dataset ($N \approx 233$k nodes) using an NVIDIA A6000 GPU. We evaluated three configurations:

1. **FP32 Baseline:** Standard single-precision inference.
2. **INT8 Standard:** W8A8 quantization without outlier mitigation.
3. **INT8 + *HD-NES*:** W8A8 quantization incorporating our proposed rotational smearing (block size = 64).

For the INT8 variants, dense matrix multiplications were executed using the `gemm-int8` CUDA kernel from IST-DASLab[2]. For the aggregation phase, we implemented a custom INT8 scatter CUDA kernel. We note that this implementation serves as a proof of concept to validate the method's feasibility; further runtime reductions are achievable with kernel fusion and optimization.

Results are detailed in Table 4. The transition from FP32 to INT8 yields a massive throughput improvement ($2.32\times$ speedup). Crucially, the injection of the ***HD-NES*** transform adds only 0.27 ms of latency, representing a marginal $\approx 1.3\%$ overhead relative to the quantized baseline.

[1] `https://pytorch.org/blog/hadacore/`

[2] `https://github.com/IST-DASLab/gemm-int8`

We further isolated the execution time of the block-wise Hadamard transform kernel, measuring 0.043 ms per pass (block size = 64). This confirms that the overhead is effectively amortized by the speedup gained from low-precision arithmetic.

Table 4: Inference latency analysis of a single GCN convolution layer on the Reddit dataset (Batch size 1, NVIDIA A6000). The introduction of *HD-NES* results in a negligible latency penalty (1.3%) compared to standard INT8 quantization, while preserving the $2.29\times$ speedup over FP32.

PRECISION	CONFIGURATION	LATENCY (MS) ↓	SPEEDUP ↑
FP32	BASELINE	47.66	1.00×
INT8	STANDARD	20.57	2.32×
INT8	+ *HD-NES* (OURS)	20.84	2.29×
INT4	STANDARD	17.052	2.79×
INT4	+ *HD-NES* (OURS)	17.234	2.77×
Hadamard Transform INT8 (Isolated)		*0.043*	–
Hadamard Transform INT4 (Isolated)		*0.031*	–

A.4 KERNEL FUSION FOR GAT EDGE SMEARING

While edge-level smearing involves $\mathcal{O}(|\mathcal{E}|)$ operations, its practical runtime is largely dictated by memory bandwidth rather than arithmetic intensity. Standard GAT implementations process message weighting, $\texttt{message}(x_j, \alpha) = \alpha \cdot x_j$, as a memory-intensive operation. A naive application of *HD-NES* would require an additional round-trip to global memory to load these messages for rotation.

To eliminate this bottleneck, we employ a Fused Message-*HD-NES* Kernel. Since the Hadamard transform is a block-wise operation requiring only local data, we fuse it directly into the message weighting pipeline.

Experimental Validation. We benchmarked this fused kernel against a standard FP32 implementation on the Reddit dataset using an NVIDIA RTX 6000 pro GPU. configuration used was a hidden size of 16 with 4 attention heads. As shown in Table 5, the fused INT8 kernel achieves a **2.52× speedup**. This confirms that the arithmetic cost of the Hadamard transform is effectively hidden by the latency savings from kernel fusion and reduced memory traffic, making edge-level smearing practically free in terms of inference latency.

Table 5: Runtime analysis of the GAT Message phase on the Reddit dataset (RTX 6000 Pro). The "Fused Kernel" performs message weighting, *HD-NES* in a single pass.

OPERATION	TIME (MS)
FP32 MESSAGE WEIGHTING (BASELINE)	41.443
INT8 FUSED KERNEL (MSG WEIGHT + *HD-NES*)	**16.765**
SPEEDUP	**2.52×**

B Implementation details

B.1 Training Setup

We relied on the Adam optimizer for all experiments. For the smaller citation networks (Cora, Citeseer, Pubmed), we utilized full-batch training.

GraphSAINT Sampling for OGB. For the large-scale datasets (`ogbn-arxiv` and `Ogbl-Citation2`), full-batch training is computationally prohibitive. We therefore employed **GraphSAINT** random walk sampling for the training phase. We configured the sampler with a walk length of 3 for both datasets. To balance coverage and training speed, we set the number of sampling roots (num setups) to 30 for `ogbn-arxiv` and 100 for `Ogbl-Citation2`. Crucially, while training relied on subgraph sampling, all validation and testing evaluations were performed on the **entire graph** (full-batch inference) to ensure the reported metrics reflect the model's performance on the complete global topology.

B.2 Model Architectures

Table 6 outlines the hyperparameters for the backbone models. For all batched experiments, we used a batch size of 128. All GAT models used 8 attention heads. All GIN architectures used 2-layer MLPs, except those for citation networks which used a single linear layer. Experiments comparing to previous methods used those methods model Architectures.

Model Arch.	**# Layers**		**# Hidden Units**		**Output MLP**	
	Citation	OGB	Citation	OGB	Citation	OGB
GCN	2	3	16	256	×	✓
GAT	2	3	8	36	×	✓
GIN	2	5	16	256	×	✓

Table 6: High level description of the architectures evaluated for citation networks and OGB.

B.3 *HD-NES* Application

Our method is selectively applied to specific layers of the GNN; for the remaining layers, we employ standard uniform quantization. The block size for the Hadamard transformation is treated as a hyperparameter and is tuned on a per-dataset basis. Table 7 details the specific insertion points of the *HD-NES* module for each architecture.

Table 7: Implementation details of the proposed method across different GNN architectures. The method is applied at two specific stages: Post-Message Passing and Post-Aggregation. Where the method is not applied, the baseline regular quantization is used.

Model	**Post-Message Passing**	**Post-Aggregation**
GIN	Layers $l \geq 2$	All Layers
GCN	Layers $l \geq 2$	Layers $l \geq 2$
GAT	All Layers	All Layers

C Ablation Studies and Analysis

In this section, we provide a detailed analysis of the hyperparameters and design choices governing *HD-NES*. All ablation experiments were conducted using the GIN architecture on the Ogbn-Arxiv dataset in the W4A4 quantization regime.

C.1 Impact of Hadamard Block Size

The block size B determines the "smearing range"; the number of nodes among which the energy of a high-degree hub is redistributed. Figure 3 (Blue bars) illustrates the impact of B on validation accuracy.

C.2 Axis of Rotation: Nodes vs. Features

Standard rotational quantization methods in LLMs often rotate the weight or activation matrices along the hidden dimension (feature axis) to make the channel distribution Gaussian. We argue that for GNNs, this approach is fundamentally flawed because the outlier source is topological (node degree).

To test this, we implemented a variation of *HD-NES* where the Hadamard transform $\mathbf{H}$ is applied to the feature dimension D ($\mathbf{XH}$) rather than the node dimension N ($\mathbf{HX}$). (Feature dimension of the GIN model is 256 so this is the max block size with feature rotations)

The results are plotted in Figure 3 (Red bars).

- ***HD-NES* (Node Axis):** Consistently improves performance over the baseline (31.3%) by flattening the degree-induced variance.
- **Feature Axis Rotation:** consistently degrades performance, dropping accuracy to $\approx 21\%$.

Analysis: High-degree nodes possess large magnitudes across *all* feature channels. Rotating features mixes these large values with each other, preserving the large norm of the node vector. Consequently, the node remains an outlier relative to the rest of the graph, and the quantization grid remains stretched.

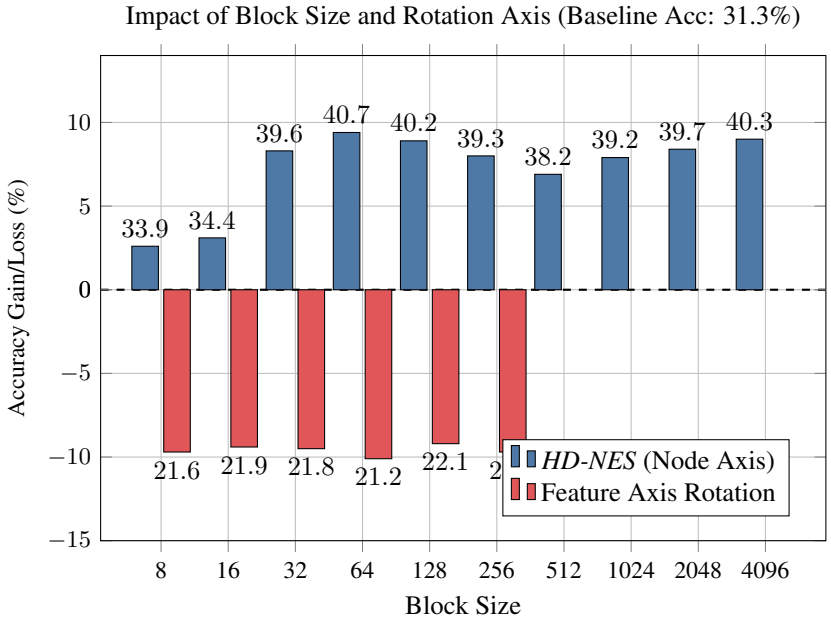


Figure 3: Ablation study on OGB-Arxiv (GIN W4A4). The dashed line represents the standard QAT baseline (31.3%). **Blue bars** denote our proposed node-axis smearing, showing peak improvement at $B = 64$. **Red bars** denote feature-axis rotation, which degrades performance by failing to address topological outliers.

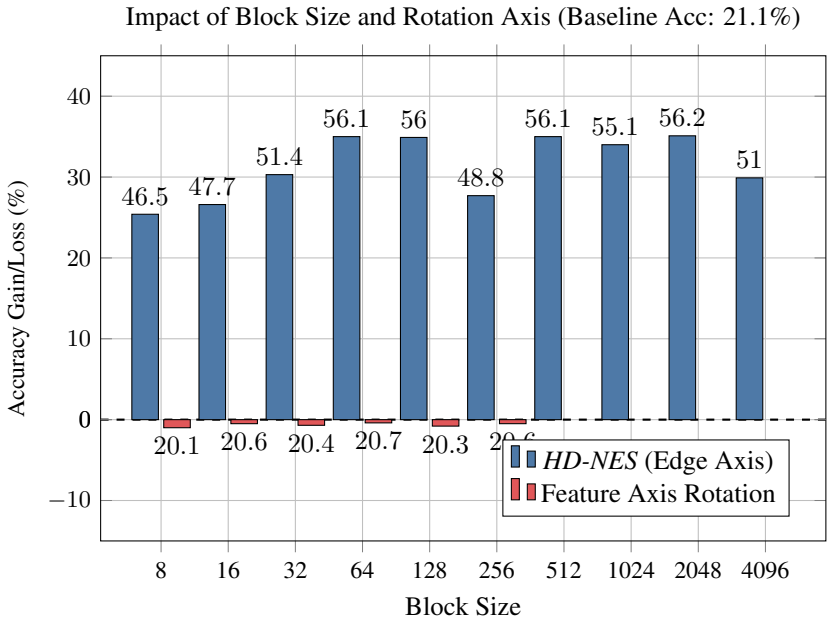


Figure 4: Ablation study on OGB-Arxiv (GatV2 W4A4). The dashed line represents the standard QAT baseline (21.1%). **Blue bars** denote our proposed edge-axis smearing, showing peak improvement at $B = 64$. **Red bars** denote feature-axis rotation, which degrades performance.

D GAT ANALYSIS

D.1 ATTENTION COEFFICIENT DISTRIBUTIONS OF GAT

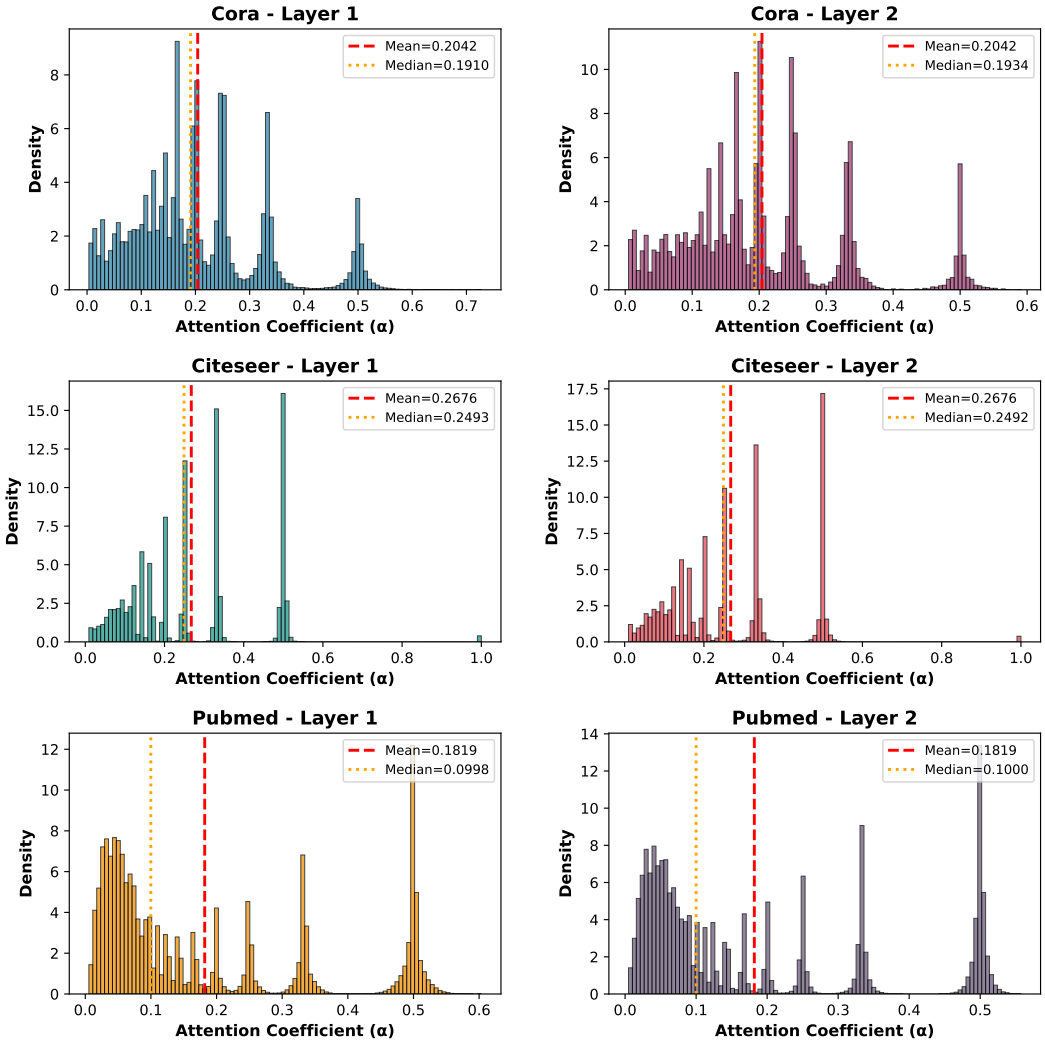


Figure 5: Attention coefficient distribution of a 2 layer GAT network trained for 200 epochs (fp32)

D.2 GAT Weighted Edge Messages Distribution

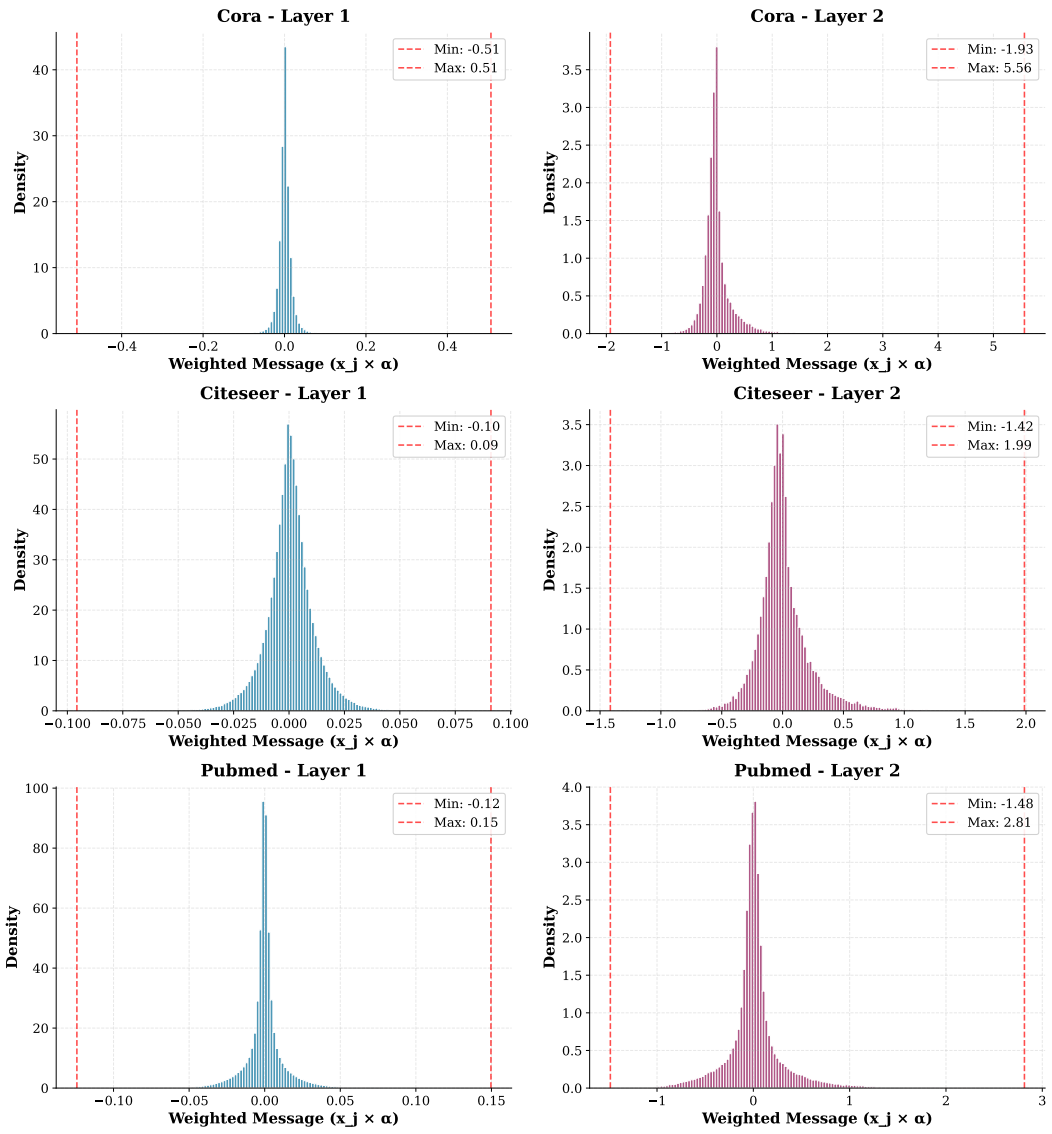


Figure 6: GAT Weighted Edge Messages Distribution of a 2 layer GAT network trained for 200 epochs (fp32)

E DATASETS

To evaluate the robustness of *HD-NES* across varying scales and topological structures, we utilize a diverse set of standard benchmarks, ranging from small citation networks to large-scale Open Graph Benchmark (OGB) datasets.

The specific characteristics of these datasets are summarized in Table 8. Crucially, these datasets exhibit power-law degree distributions, where a minority of hub nodes possess degrees significantly higher than the average. As discussed in Section 3.1, these hubs are the primary source of activation outliers.

- **Citation Networks (Cora, Citeseer, Pubmed):** Standard benchmarks for semi-supervised node classification. Nodes represent documents and edges represent citation links.
- **OGB-Arxiv:** A directed graph representing the citation network between all Computer Science arXiv papers indexed by MAG. This dataset tests the scalability of the method on a significantly larger graph structure.
- **Ogbl-Citation2:** A link prediction dataset aimed at predicting missing citations in a collaborative network.
- **Reddit Binary:** A social network dataset used for graph classification. It consists of graphs corresponding to online discussion threads, where nodes are users and edges represent responses.

Table 8: Summary of dataset statistics used in experiments. Note that for Reddit Binary (Graph Classification), the node and edge counts represent the average per graph.

Dataset	Task	Nodes	Edges	Features	Classes
Node Classification & Link Prediction					
Cora	Node Classif.	2,708	5,429	1,433	7
Citeseer	Node Classif.	3,327	4,732	3,703	6
Pubmed	Node Classif.	19,717	44,338	500	3
OGB-Arxiv	Node Classif.	169,343	1,166,243	128	40
Ogbl-Citation2	Link Pred.	2,927,963	30,561,187	128	–
Graph Classification					
Reddit Binary	Graph Classif.	∼429	∼497	–	2

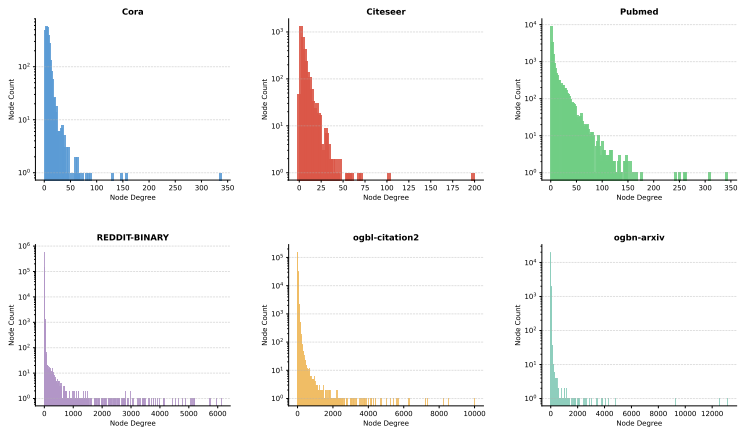


Figure 7: Degree distribution of the datasets used in evaluation. The plot confirms the power-law property, where high-degree hubs (the right tail) are statistically distinct from the majority of low-degree nodes. These topological outliers drive the quantization range explosion targeted by *HD-NES*.

Compute scaling vs. model size. Figure 8 highlights the mismatch between parameter count and inference cost in GNNs. In contrast to Transformer-style scaling trends, a small GCN operating on a large graph can require comparable or greater forward-pass compute because message passing scales with graph structure rather than model size alone.

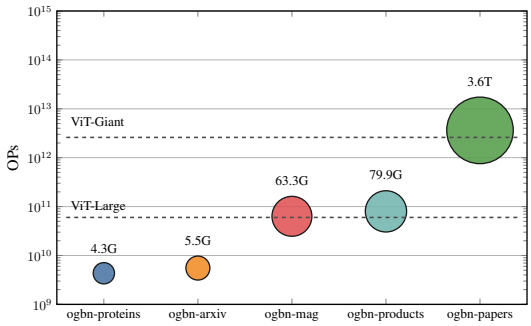


Figure 8: GNN computational cost vs. model size Despite having orders of magnitude fewer parameters than large vision transformers, GCNs on large graphs require comparable computational cost due to message-passing overhead on graph structure. Circle diameter represents node count.

F THE EXPRESSIVITY–QUANTIZATION TRADE-OFF

To maximize structural expressivity, message-passing GNNs typically rely on injective, magnitude-expanding aggregation functions like sum-pooling, which aligns with the discriminative power of the Weisfeiler-Lehman (1-WL) test Xu et al. (2019). However, preserving this multiset cardinality in the continuous domain inherently induces activation magnitudes that scale linearly with node degree (Section 3.1), creating a severe bottleneck when transitioning to low-precision discrete representations.

The fundamental challenge lies in how standard uniform quantization translates this degree-induced range explosion into a *resolution blind spot*. To prevent clipping the extreme outliers generated by hub nodes, the quantization step size S must be scaled up drastically. When S becomes larger than the continuous feature variations that distinguish different structural neighborhoods, these distinct representations are mapped to the exact same discrete integer. We term this loss of distinguishability *Quantization Bin Collapse*. Consequently, standard low-bit quantization actively degrades the very injectivity that sum-aggregation aims to preserve.

A natural question is whether we can avoid this hub-induced range explosion entirely by replacing sum-aggregation with mean-pooling or strong degree normalization. However, this approach does not solve the expressivity problem; it simply circumvents it by creating an *architectural blind spot*. Mean-pooling fundamentally discards structural density (multiset cardinality) before quantization even occurs. It fails to serve as a viable alternative because it abandons 1-WL expressivity by design, which causes the model to collapse to random-guessing baselines on structural tasks (Table 9).

The dilemma is therefore between an architectural blind spot (mean aggregation) and a resolution blind spot (quantized sum aggregation). We acknowledge that discrete quantization inherently cannot perfectly preserve continuous 1-WL expressivity, and *HD-NES* does not claim to eliminate this resolution blind spot. Instead, *HD-NES* serves to significantly *mitigate* it. By compressing the global dynamic range and shrinking the required step size S, our method mitigates bin collapse. This allows the network to preserve the expressivity advantages of sum-aggregation without resorting to the architectural sacrifice inherent in mean-pooling, maintaining a strict performance advantage even under 4-bit constraints (Table 9).

Table 9: Comparison of Sum Aggregation (GIN-0) and Mean Aggregation on graph classification tasks. Results denote Accuracy (%) $\pm$ Std Dev. The data highlights that sum aggregation maintains its expressivity advantage over mean aggregation under 4-bit quantization, particularly when enhanced by ***HD-NES***.

Dataset	**Aggregation Model**	**FP32**	**Int 4**	***HD-NES* Int 4**
REDDIT-BINARY	GIN-0 (Sum)	92.1 ± 2.2	83.3 ± 0.9	$\mathbf{86.7 \pm 1.2}$
	Mean 1 layer	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
REDDIT-MULTI5K	GIN-0 (Sum)	52.3 ± 2.8	48.4 ± 1.3	$\mathbf{51.2 \pm 1.1}$
	Mean 1 layer	20.0 ± 0.0	20.0 ± 0.0	20.0 ± 0.0

G PERFORMANCE ON HETEROPHILIC GRAPHS

To evaluate the robustness of *HD-NES* beyond standard homophilic benchmarks, we extend our analysis to different graph topologies, specifically evaluating on the heterophilic Chameleon and Squirrel datasets. For this evaluation, we follow the experimental protocol introduced by Platonov et al. Platonov et al. (2023).

As shown in Table 10, standard uniform quantization struggles to maintain accuracy on these datasets, especially in lower bit-width regimes. In contrast, applying *HD-NES* redistributes the energy effectively, yielding consistent accuracy gains over the standard baseline for both 8-bit and 4-bit integer quantization.

Table 10: Comparison of Standard quantization and *HD-NES* on heterophilic graphs (Chameleon and Squirrel) across different bit-widths. Results denote Accuracy (%).

Dataset	**Bit-width**	**Standard**	**HD-NES**
Chameleon	FP32	40.96	-
	INT8	39.31	**40.48**
	INT4	39.20	**39.64**
Squirrel	FP32	39.27	-
	INT8	37.93	**39.12**
	INT4	36.60	**37.26**

H PROOF OF THEOREM 4.2

Proof. The goal of the randomized Hadamard transform is to reduce the *peak magnitude* of any vector before quantization. Let $\mathbf{z} = \mathbf{H}_B\mathbf{D}\mathbf{x}$. We prove that, with high probability,

$$\|\mathbf{z}\|_\infty = \max_{1\le i\le B} |z_i|$$

is small (of order $\|\mathbf{x}\|_2/\sqrt{B}$), which directly controls the dynamic range of the quantized values.

1. Component Expansion. Let $\mathbf{D} = \mathrm{diag}(\epsilon_1, \dots, \epsilon_B)$ where $\epsilon_j \in \{+1, -1\}$ are i.i.d. Rademacher random variables. The i-th component is

$$z_i = \sum_{j=1}^{B} (\mathbf{H}_B)_{ij}\epsilon_j x_j.$$

Since $\mathbf{H}_B$ is normalized, $|(\mathbf{H}_B)_{ij}| = B^{-1/2}$.

2. Moment Generating Function (MGF) Bound. Following a Chernoff-type argument, consider the MGF of z_i:

$$\mathbb{E}[e^{\lambda B z_i}] = \prod_{j=1}^{B} \mathbb{E}[e^{\lambda B(\mathbf{H}_B)_{ij}\epsilon_j x_j}] = \prod_{j=1}^{B} \cosh(\lambda\sqrt{B}\, x_j) \le \prod_{j=1}^{B} e^{\lambda^2 B x_j^2/2} = e^{\lambda^2 B\|\mathbf{x}\|_2^2/2},$$

where the inequality uses $\cosh(u) \le e^{u^2/2}$.

3. Chernoff Bound. By Markov's inequality, for any $\lambda > 0$,

$$\Pr[|z_i| \ge t] \le 2\,\frac{\mathbb{E}[e^{\lambda B z_i}]}{e^{\lambda B t}} \le 2\exp\left(\frac{\lambda^2 B\|\mathbf{x}\|_2^2}{2} - \lambda B t\right).$$

The exponent is minimized at $\lambda = t/\|\mathbf{x}\|_2^2$, yielding

$$\Pr[|z_i| \ge t] \le 2\exp\left(-\frac{t^2 B}{2\|\mathbf{x}\|_2^2}\right).$$

4. Union Bound. Applying the union bound over all B coordinates,

$$\Pr[\|\mathbf{z}\|_\infty \ge t] \le 2B\exp\left(-\frac{t^2 B}{2\|\mathbf{x}\|_2^2}\right).$$

5. Solving for the Threshold. Set the right-hand side equal to δ:

$$2B\exp\left(-\frac{t^2 B}{2\|\mathbf{x}\|_2^2}\right) = \delta \implies \exp\left(-\frac{t^2 B}{2\|\mathbf{x}\|_2^2}\right) = \frac{\delta}{2B}.$$

Taking the natural logarithm and rearranging gives

$$t = \|\mathbf{x}\|_2\sqrt{\frac{2\log(2B/\delta)}{B}}.$$

Thus, with probability at least $1-\delta$,

$$\|\mathbf{z}\|_\infty \le \sqrt{\frac{2\log(2B/\delta)}{B}}\,\|\mathbf{x}\|_2,$$

which completes the proof. □

I PROOF OF LEMMA 4.1

Proof. Since $\Psi = \mathbf{H}_B\mathbf{D}$ is orthogonal:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 = \|\Psi^{-1}\mathbf{z} - \Psi^{-1}\hat{\mathbf{z}}\|_2^2 = \|\Psi^{-1}(\mathbf{z} - \hat{\mathbf{z}})\|_2^2 = \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2,$$

where the last equality follows from the norm-preserving property of orthogonal transformations.

Let $e_i = z_i - \hat{z}_i$ denote the quantization error in coordinate i. Under the standard high-resolution quantization model, the coordinate-wise error is treated as a zero-mean random variable with

$$\mathbb{E}[e_i^2] \approx \frac{S^2}{12},$$

where S is the quantization step size. Therefore, by linearity of expectation,

$$\mathbb{E}\big[\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2\big] = \mathbb{E}\left[\sum_{i=1}^{B} e_i^2\right] = \sum_{i=1}^{B} \mathbb{E}[e_i^2] \approx \frac{B\,S^2}{12}.$$

Substituting $S = 2\|\mathbf{z}\|_\infty/(2^b - 1)$, we obtain

$$\mathbb{E}\big[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\big] \approx \frac{B}{12} \cdot \frac{4\|\mathbf{z}\|_\infty^2}{(2^b - 1)^2} = \frac{B\,\|\mathbf{z}\|_\infty^2}{3(2^b - 1)^2}.$$

Hence, for fixed block size B and bit-width b, the expected reconstruction error is proportional to $\|\mathbf{z}\|_\infty^2$ under the high-resolution quantization model. □

Remark I.1 (Worst-case bound). The above result relies on the high-resolution quantization model. A deterministic bound holds without any distributional assumption: since $|e_i| \le S/2$ for every coordinate, we have $e_i^2 \le S^2/4$, and therefore

$$\|\mathbf{z} - \hat{\mathbf{z}}\|_2^2 = \sum_{i=1}^{B} e_i^2 \le \frac{B\,S^2}{4} = \frac{B\,\|\mathbf{z}\|_\infty^2}{(2^b - 1)^2}.$$

This worst-case upper bound, valid at any bit-width, shows that the reconstruction error scales as $\mathcal{O}(\|\mathbf{z}\|_\infty^2)$ regardless of the noise model.

J EXTERNAL CODE, SOFTWARE, AND BASELINE IMPLEMENTATIONS

This appendix is provided to document the external assets used in this work, including datasets, codebases, and benchmark implementations.

Table 11: External codebases, libraries, and baseline implementations used by the project.

Asset	How used in this paper	Citation	Version / commit	License
PyTorch Geometric (PyG)	Graph learning library used for GCN, GIN, GAT/GATv2 layers, graph data handling, and message-passing primitives	Fey & Lenssen (2019)	Version – `[2.7.0]`	MIT license
Open Graph Benchmark (OGB)	Dataset loaders, standardized splits, and evaluation protocols for `ogbn-arxiv` and `Ogbl-Citation2`	Hu et al. (2020)	Version – `[1.3.6]`	MIT license
GraphSAINT sampling	Random-walk subgraph sampler used for training on large-scale OGB graphs, as described in Section B	Zeng et al. (2019)	Via PyG	MIT license
`hadacore`	Starting point for the Hadamard transform kernel; adapted to support block-wise left Hadamard transforms for *HD-NES*	had	Commit – `[2391954]`	BSD-3
`gemm-int8`	INT8 dense matrix multiplication CUDA kernel used in the runtime benchmarks in Section A	Castro et al. (2024)	Commit – `[27f779c]`	MIT license
DegreeQuant	Quantization-aware GNN baseline; *HD-NES* is integrated with DegreeQuant in Table 2	Tailor et al. (2021)	Commit – `[257d6bc]`	BSD-3
SMP	Smoothness-aware message propagation baseline; *HD-NES* is integrated with SMP in Table 3	Wang et al. (2023)	Commit – `[00dd14c]`	no explicit license found