

Improving Reliability in Quantized Graph Neural Networks with Node-Wise Entropy-driven Temperature Scaling

Hadi Mousanejad Jeddi

Department of Electrical Engineering
 Linköping University
 Linköping, Sweden
 hadi.mousanejad.jeddi@liu.se

Jose Nunez-Yanez

Department of Electrical Engineering
 Linköping University
 Linköping, Sweden
 jose.nunez-yanez@liu.se

Abstract—Graph Neural Networks (GNNs) are one of the most powerful learning methods for graph-structured data and their quantization significantly reduces memory and computational requirements on edge devices. In this paper, we show that the quantization of node features, edge connections, and hidden representations degrades confidence calibration. To address this issue we propose a node-wise temperature scaling method that dynamically calibrates model confidence by aggregating entropy-based uncertainty from graph-structured data. Our approach combines self-entropy, neighborhood-entropy, and shortest-path distances to labeled nodes into a unified feature representation followed by a learnable transformation to compute temperature values for each node. We integrate and evaluate our approach using a dataflow hardware accelerator optimized for multi-precision GNN models, which supports efficient training and inference on-device. Our method significantly improves calibration by reducing Expected Calibration Error (ECE) and Negative Log-Likelihood (NLL) by up to 95% and 66%, respectively, across multiple datasets, without decreasing accuracy after quantization. The implementation is publicly available.¹

Index Terms—FPGA, calibration, graph neural network, hardware accelerator, quantization, temperature scaling

I. INTRODUCTION

Graph Neural Networks (GNNs) are increasingly used to analyze graph-structured data in multiple real-world applications [1]. They have been effectively deployed in numerous areas, such as social network analysis, urban traffic flow modeling, recommendation systems, and molecular structure understanding. Despite their effectiveness, small GNN models can still demand significant computational resources due to the large input graph size. This has led to the development of quantized GNNs due to their negligible accuracy loss and improved efficiency in memory and computation requirements, particularly on edge devices [2]. Recent studies indicate that modern neural networks are often not well-calibrated and the output values cannot be used as confidence estimates. In particular, quantization can worsen model miscalibration by reducing numerical precision, which distorts confidence estimates and inflates overconfidence in incorrect predictions [3]. Especially due to limited resources, edge devices are more vulnerable to such overconfident yet incorrect predic-

tions. Moreover, we observe that quantization can significantly worsen miscalibration as shown in Figure 1.

Calibration in machine learning measures how well a model’s predicted confidence matches the actual likelihood of correctness, which is crucial for improving its trustworthiness. For example, a well-calibrated model that outputs 80% confidence should make correct predictions 80% of the time [4]. Recent studies show that measuring calibration error at the level of individual samples is inherently difficult, especially in graph-based models where node predictions are structurally dependent. This may result in both under-confident and over-confident predictions, even if prediction accuracy remains high, which can lead to unsafe or unreliable decision-making in high-risk areas such as self-driving cars [5], [6]. Although numerous calibration techniques for GNNs have been proposed, they are largely tailored to full-precision environments and often depend on auxiliary models or complex ensemble techniques, which hinder their use in ultra-low-bit edge deployment. Therefore, achieving reliable confidence calibration for quantized GNNs in resource-limited environments remains largely unresolved.

In this work, we propose a novel confidence calibration method for quantized Graph Neural Networks (GCNs) deployed on edge devices. Our approach, Node-wise Entropy-driven Temperature Scaling (NHTS), computes a unique softmax temperature for each node based on self-entropy, neighborhood-entropy, node degree, and distance to labeled nodes, capturing both local uncertainty and graph structure. To support low-bit quantized training, we integrate NHTS into the hardware-aware training and quantization pipeline proposed in the SGRACE, a scalable graph convolutional and attention network framework for efficient GNN deployment on edge devices [7]. NHTS dynamically determines optimal clipping ranges to reduce information loss prior to quantization to SGRACE. This enables efficient 8/4/2-bit GCN deployment without sacrificing accuracy. Experiments on standard benchmarks show that our method consistently reduces Expected Calibration Error (ECE) and Negative Log-Likelihood (NLL) across various bit-widths, while maintaining classification accuracy. Our main contributions are:

- We develop a clipping-optimized dynamic quantization

¹<https://github.com/hadimsnj/NHTS>

process and integrate it into the hardware-aware quantized training pipeline of SGRACE to enable efficient low-bit GCNs without significant accuracy degradation.

- We propose Node-wise Entropy-driven Temperature Scaling (NHTS), a novel calibration technique that jointly optimizes quantization and confidence calibration by adaptively adjusting node-wise softmax temperatures based on entropy and graph structural features.
- We evaluate NHTS on standard benchmarks, where it improves calibration with up to 95% lower ECE and 66% lower NLL at 2-bit precision, while also increasing accuracy.

II. BACKGROUND

A. Quantized GNNs and Specialized Hardware Accelerators

Although GNNs often have small parameter sizes, their irregular memory access patterns, sparse data layouts, and the high computational cost of graph aggregation remain challenging for hardware deployment [2]. For instance, a GCN with 81KB of parameters may require 19 GigaOPs to process a large graph. To address these demands, quantization techniques have been adopted to reduce both computational and memory overhead [8]. Quantized GNNs use low-bit representation formats, such as 8, 4 and 2-bit, to improve energy efficiency and increase throughput.

Several quantization strategies have been proposed. Degree-Quant (DQ) [8] uses stochastic masking to retain full precision for high-degree nodes combined with quantization-aware training (QAT). A2Q [9] learns node-wise bit-widths and step sizes through local gradients to implement mixed-precision quantization. Some approaches have focused on implementing GNNs and their quantized forms on FPGAs. For instance, several specialized accelerators, such as HyGCN [10], AWB-GCN [11], GCNAX [12], and gFADES [13], have been developed to optimize dataflows, memory hierarchies, and sparsity patterns, which are critical for improving throughput with hardware-constrained devices. QEGCN [14], utilizes INT8 and INT4 quantization along with pipelining and edge-level parallelism to accelerate GCNs.

On the other hand, SGRACE offers a hardware/software co-designed framework for efficient GNN training and inference on FPGAs. It supports quantization from 1 to 8 bits through a Hardware-Aware Quantized Training scheme that executes forward passes directly on quantized hardware (see Figure 2). This reduces the mismatch between training and inference and maintains accuracy under low-precision constraints. Its fully on-device workflow makes it especially suited for applying calibration methods to quantized GNNs in resource-constrained environments. In our implementation, we utilize only the GCN part of SGRACE and exclude attention mechanisms.

B. Model Calibration in Deep Learning and GNNs

Temperature Scaling (TS) is a popular calibration method that is known for its simplicity, data efficiency, and ability to preserve accuracy. In addition to TS, methods like Attended

Temperature Scaling (ATS) [15], Vector Scaling (VS) [4], and Ensemble Temperature Scaling (ETS) [16] offer more robust calibration under noisy or imbalanced conditions.

However, these methods are typically designed for i.i.d. data and are less effective on graph-structured data due to node dependencies. In GNNs, node confidence can be affected by node-specific factors such as position, degree, and distance to labeled nodes. To solve this, several calibration methods have been proposed. CaGCN [5] learns node-specific temperatures by using a second GCN that considers the graph structure. GATS [6] introduces attention mechanisms based on prediction confidence, neighbor similarity, and label distance. GETS [17] uses a mixture-of-experts framework that leverages logits, node features, and degree information.

While these methods have improved calibration in GNNs, they depend on complex algorithms, such as second GNN, attention or mixture-of-experts mechanisms, which limit hardware efficiency. Moreover, they are mainly designed for full-precision GNNs, and their behavior under quantization remains largely unexplored. Prior studies in CNNs [18], [3] have shown that quantization often degrades calibration, particularly at lower bit-widths. In this work, we focus on calibration in quantized GNNs and propose a lightweight solution tailored for efficient deployment.

III. DEFINITIONS

Expected Calibration Error (ECE) [4] evaluates the difference between a model’s predicted confidence and its actual accuracy. To compute ECE, predictions are grouped into M confidence bins. For each bin B_m , the average accuracy $\text{acc}(B_m)$ and average confidence $\text{conf}(B_m)$ are computed. The ECE is calculated using the following formula:

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (1)$$

Where n is the total number of samples, and B_m refers to the group of predictions with confidence scores within the m -th bin. Based on the formula, lower ECE values indicate higher reliability of confidence estimates.

Negative log likelihood (NLL) [4] measures how well a model’s predicted probabilities align with the true labels. For a given dataset of n samples, NLL is defined as:

$$\text{NLL} = - \sum_{i=1}^n \log \hat{\pi}(y_i | x_i), \quad (2)$$

where $\hat{\pi}(y_i | x_i)$ is the predicted probability assigned to the true class y_i given input x_i . Models with low NLL better estimate the true label probabilities.

Entropy[19] helps to evaluate how well a model’s confidence aligns with its actual accuracy and is defined as:

$$H(\mathbf{p}) = - \sum_{i=1}^N p_i \log p_i, \quad (3)$$

where $\mathbf{p} = [p_1, \dots, p_N]$ is the softmax output over N classes. Over-confident models typically exhibit low entropy,

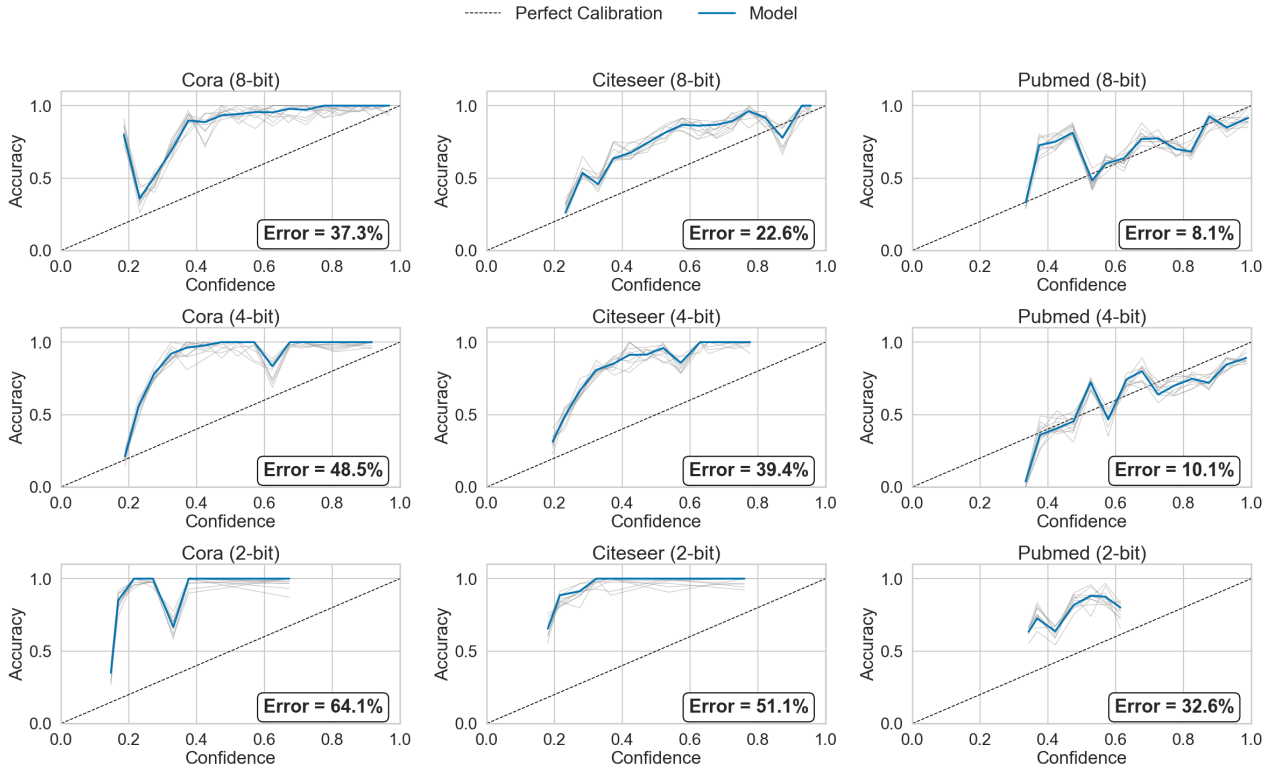


Fig. 1. Comparing model confidence vs. accuracy for 8-bit, 4-bit, and 2-bit quantized models on Cora, Citeseer, and Pubmed datasets.

while under-confident models display high entropy. Moreover, entropy is a useful tool for adjusting model confidence to improve reliability.

Temperature Scaling [4] is a post-hoc calibration technique that rescales logits before the softmax layer using a single scalar temperature parameter $T > 0$. For a logit vector z , the softmax function is adjusted as follows:

$$\sigma(z/T)_k = \frac{e^{z_k/T}}{\sum_{j=1}^K e^{z_j/T}}, \quad (4)$$

where K is the number of classes. Higher values of T produce softer probability distributions and help reduce over-confidence, while lower values of T sharpen the distribution and can address under-confidence. For clarity, Table I provides a list of all the symbols and variables used in the paper.

IV. METHODOLOGY

In this section, we present our node-wise entropy-driven temperature scaling method for confidence calibration in quantized GCNs, which are trained and calibrated on edge devices. We first describe the pre-quantization process and hardware-aware quantized training using the SGRACE accelerator, and then detail our node-wise temperature scaling approach.

A. Quantization Strategy and On-Device Training with SGRACE

An essential step in achieving well-calibrated confidence estimates in quantized neural networks, especially when run-

TABLE I
SUMMARY OF VARIABLES USED IN THE NHTS ALGORITHM FOR QUANTIZED GNN CALIBRATION.

Variable	Description
x_i	Input tensor elements
α	Clipping threshold for quantization, selected to minimize information loss based on data distribution (e.g., max, MSE, or KLD criteria)
$Q_\alpha(x)$	Quantized value of x using threshold α
$H(p)$	Entropy of softmax distribution p
\hat{H}_i	Self-entropy of node i
$\hat{H}_i^{(N)}$	Neighborhood entropy of node i
D_i	Degree of node i
S_i	Shortest-path to nearest labeled node
ϕ_i	Feature vector: $[\hat{H}_i, \hat{H}_i^{(N)}, D_i, S_i]$
T	Temperature scalar (node-specific in NHTS)
w_H, b	Learnable parameters for temperature computation

ning on edge devices, is to effectively handle outlier data by ensuring an optimal clipping range before quantization. A well-chosen clipping range ensures that quantization retains most of the information, which is beneficial for confidence calibration estimation, whereas poor clipping can cause significant accuracy degradation and make later calibration methods ineffective. In the pre-quantization step, we perform a dynamic range analysis (DRA) to determine clipping thresholds tailored to the data distribution of each tensor. We evaluate three standard clipping strategies that are widely used in prior quantization studies to define the symmetric clipping range

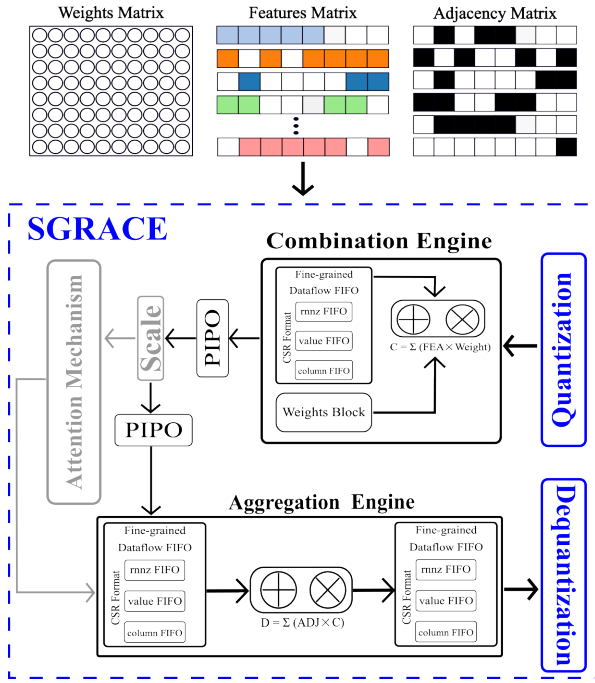


Fig. 2. SGRACE dataflow design supports efficient training and inference of quantized GNNs on edge hardware [7].

$[-\alpha, \alpha]$:

- *Max-Absolute Clipping*:

$$\alpha_{\max} = \max_i |x_i|, \quad (5)$$

- *MSE Minimization Clipping*:

$$\alpha_{\text{mse}} = \arg \min_{\alpha} \mathbb{E} [(x - Q_{\alpha}(x))^2], \quad (6)$$

- *KLD Minimization Clipping*:

$$\alpha_{\text{kld}} = \arg \min_{\alpha} D_{\text{KL}}(P(x) \| P(Q_{\alpha}(x))), \quad (7)$$

where $Q_{\alpha}(x)$ denotes quantization using range α , and $P(\cdot)$ indicates the distribution. To minimize the quantization error, we dynamically determine the optimal threshold instead of using a predetermined method:

$$\alpha^* = \arg \min_{\alpha \in \{\alpha_{\max}, \alpha_{\text{mse}}, \alpha_{\text{kld}}\}} L_{\text{clip}}(\alpha), \quad (8)$$

where $L_{\text{clip}}(\alpha)$ measures the quantization loss. The clipping method is selected individually for each dataset and quantization bit-width to ensure it aligns with the unique characteristics of the data and precision requirements. This adaptive strategy allows the clipping threshold to be optimally calibrated for each distribution and effectively minimizes information loss prior to quantization. Following this pre-quantization step, we perform hardware-aware quantized training of the GCN model using the SGRACE accelerator. The model is trained with 8, 4, and 2-bit precision, and the results show that accuracy remains consistently stable across different bit-widths.

B. Node-Wise Entropy-Driven Temperature Scaling for Confidence Calibration

After training and quantization, we apply confidence calibration using our proposed NHTS method. Unlike global temperature scaling, which employs a single temperature parameter across all samples, or more complex approaches such as secondary GNNs, complex attention mechanisms, and mixture-of-experts models that add millions of parameters and require an additional forward pass, our method relies only on simple per-node features. Recent work on entropy-based calibration [19] uses the entropy of softmax probabilities as a measure of uncertainty. However, this measure alone is insufficient in GNNs, where each node aggregates information from its neighbors. As a result, a misclassified node regardless of its confidence level can cause error propagation throughout the neighborhood. Moreover, a node's calibration correlates with the behavior of its neighbors [6], where nodes that are closer to labeled nodes tend to receive stronger label influence and better predictions, while nodes that are farther away may receive noisy signals. In order to measure both uncertainty and structural information, firstly, we compute a node's self-entropy and the neighborhood entropy to quantify predictive uncertainty. Secondly, we incorporate the shortest-path distance to the closest labeled node to reflect the strength of label influence. These features are combined into the following feature vector:

- 1) **Self-Entropy**: Measures uncertainty of the model for node

$$\hat{H}_i = -\frac{1}{\log C} \sum_{c=1}^C p_{ic} \cdot \log(p_{ic}), \quad (9)$$

where p_{ic} is the softmax probability for class c at node i .

- 2) **Neighborhood Entropy**: Aggregating uncertainty from the immediate neighborhood

$$\hat{H}_i^{(N)} = \frac{1}{\text{deg}(i)} \sum_{j \in \mathcal{N}(i)} \hat{H}_j, \quad (10)$$

- 3) **Node Degree**: Indicating the amount of information a node receives during message passing.

$$D_i = \text{deg}(i), \quad (11)$$

- 4) **Shortest-Path Distance to Labeled Nodes**: Determines the direct impact of ground-truth label influence on each node

$$S_i = \min_{j \in \mathcal{T}} \text{dist}(i, j), \quad (12)$$

where \mathcal{T} is the set of labeled nodes.

We combine these four components into the feature vector ϕ_i , which reflects both the uncertainty in predictions and the node's structural role. We then compute the node-specific temperature using a nonlinear transformation of this vector.

$$\phi_i = \left[\underbrace{\hat{H}_i}_{\text{self-entropy}}, \underbrace{\hat{H}_i^{(N)}}_{\text{neigh. entropy}}, \underbrace{D_i}_{\text{degree}}, \underbrace{S_i}_{\text{dist. to labels}} \right], \quad (13)$$

TABLE II
ACCURACY, ECE, AND NLL RESULTS ACROSS DATASETS AND BIT-WIDTHS.

Dataset	Bits	Metric	NoCal	NoCal+DRA	TS	ETS	VS	LTS	HTS	HnLTS	CaGCN	NHTS
Cora	8	Accuracy	83.30	84.80	84.80	84.80	84.90	84.80	84.80	84.80	84.80	84.80
		ECE	37.29	5.01	4.62	4.71	5.13	4.18	7.66	4.23	4.15	2.78
		NLL	0.941	0.515	0.522	0.515	0.516	0.528	1.339	0.635	0.522	0.500
	4	Accuracy	80.00	85.50	85.50	85.50	84.90	85.50	85.50	85.50	85.50	85.50
		ECE	48.54	17.79	9.82	3.74	4.40	3.99	4.59	6.57	4.05	2.81
		NLL	1.270	0.611	0.535	0.493	0.517	0.496	0.616	0.544	0.556	0.484
	2	Accuracy	81.50	81.90	81.90	81.90	82.70	81.90	81.90	81.90	81.90	81.90
		ECE	64.13	12.06	6.65	3.03	7.32	3.90	5.36	6.16	6.54	2.63
		NLL	1.77	0.656	0.617	0.607	0.618	0.621	0.769	0.685	0.646	0.601
Citeseer	8	Accuracy	73.40	76.90	76.90	76.90	77.00	76.90	76.90	76.90	76.90	76.90
		ECE	22.46	23.02	7.22	13.13	6.00	3.65	5.50	7.08	6.56	2.63
		NLL	0.99	0.894	0.757	0.796	0.772	0.744	0.768	0.762	0.791	0.740
	4	Accuracy	71.00	77.40	77.40	77.40	77.20	77.40	77.40	77.40	77.40	77.40
		ECE	39.40	19.86	5.34	10.41	4.33	3.82	6.47	7.20	6.65	3.29
		NLL	1.309	0.865	0.761	0.785	0.775	0.757	0.828	0.782	0.815	0.744
	2	Accuracy	70.20	70.90	70.90	70.90	71.50	70.90	70.90	70.90	70.90	70.90
		ECE	51.06	10.01	7.27	7.05	8.73	6.37	10.95	9.29	11.09	6.07
		NLL	1.680	0.927	0.914	0.914	0.910	0.902	0.983	1.888	1.014	0.911
Pubmed	8	Accuracy	83.60	84.10	85.30	84.30	84.70	85.30	84.10	85.30	84.60	85.00
		ECE	8.31	8.05	8.30	6.64	6.84	9.48	6.59	8.71	8.97	7.96
		NLL	0.686	0.606	0.498	0.458	0.469	0.470	0.473	0.456	0.476	0.455
	4	Accuracy	72.80	82.40	82.80	82.60	83.20	81.90	82.30	81.70	82.70	81.90
		ECE	13.10	8.18	8.00	7.44	6.03	9.34	6.09	5.20	8.25	3.79
		NLL	0.859	0.509	0.509	0.506	0.484	0.519	0.489	0.503	0.500	0.472
	2	Accuracy	70.70	78.60	79.50	78.70	78.80	79.40	79.60	79.10	79.40	78.60
		ECE	32.59	16.58	14.09	15.39	13.79	12.42	7.41	6.67	9.93	2.62
		NLL	1.009	0.663	0.630	0.644	0.629	0.606	0.577	0.559	0.593	0.542

Unlike traditional global temperature scaling methods that apply a uniform temperature across all samples, our method allows each node to receive a node-specific temperature based on self-entropy, neighborhood distance, node degree, and distance to training nodes. Finally, a combination function (e.g., mean, max, min, median, or sum) is applied to the vector ϕ_n to generate an overall uncertainty measure c_n , which is used to compute the temperature τ_n for node n .

$$c_n \leftarrow \text{aggregate}(\phi_n), \quad (14)$$

$$\tau_n \leftarrow \text{clamp}(\log(1 + \exp(w_H \cdot \log(c_n) + b))), \quad (15)$$

Combination function modes represent different calibration strategies, which differ based on how much correction they apply to uncertain predictions, and are selected based on the characteristics of the datasets. For example, max responds strongly to the highest uncertainty, while mean provides a balanced adjustment by averaging all inputs. By adapting to each node’s features, this formulation corrects node-level miscalibration through a systematic and gradient-friendly method.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate the performance of the proposed NHTS method across various datasets and quantization bit-widths. The analysis focuses on both prediction accuracy and calibration metrics such as ECE and NLL. All experiments are conducted on a hardware-constrained edge device to validate practical capabilities.

A. Experimental setup

All experiments were performed on the Ultra96V2 FPGA board with the PYNQ 3.1 image. The complete workflow, including training, quantization, and calibration, was implemented using the SGRACE framework, which was developed with High-Level Synthesis (HLS) for FPGA deployment. The results show that reliable prediction confidence can be maintained across multiple quantization levels, even when on-device learning is performed with limited resources.

B. DRA Performance Analysis for Low-Bit Quantization

Our experimental results indicate that Dynamic Range Analysis (DRA) not only improves accuracy but also significantly reduces ECE. In addition, DRA improves accuracy by reducing quantization noise caused by outlier data in the clipping process. This improvement is most noticeable at low-bit precision, where minor errors can lead to significant performance drops. As shown in Table II, without any range analysis, the baseline model (NoCalib) exhibits significant accuracy degradation. For instance, in 4-bit quantization on the PubMed dataset, ECE decreases by more than 35% without any calibration, while accuracy increases by approximately 10%. Similarly, on the Cora dataset with 4-bit precision, ECE reduces by approximately 60%, and NLL improves by around 52%. Moreover, in the highly constrained 2-bit quantization setting on the PubMed dataset, DRA considerably improves calibration by reducing ECE from 32.59 to 16.58, which represents a reduction of nearly 50%, and at the same time increases accuracy from 70.70% to 78.60%. These

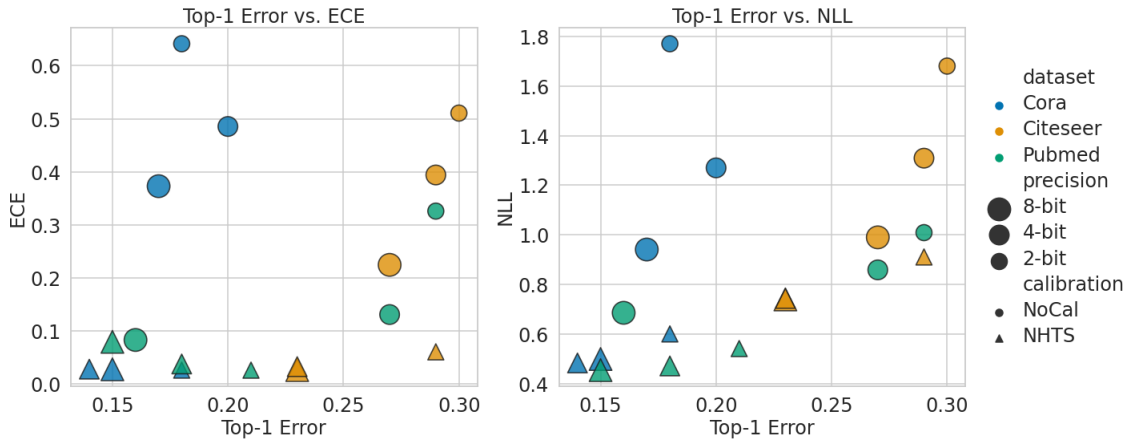


Fig. 3. Top-1 error vs. ECE and NLL across datasets and bit-widths. NHTS (triangles) performs better than the uncalibrated baseline (circles), with lower error and better calibration, especially at 4-bit and 2-bit precision.

results demonstrate that hardware-aware clipping optimizes the information maintained during quantization, which directly enhances confidence calibration.

C. Evaluation of Calibration Techniques

To eliminate the effects of quantization inconsistencies, all calibration methods are evaluated after applying DRA, which ensures that any observed performance improvements result only from the calibration methods. All competing methods, including TS, VS [4], ETS [16], as well as LTS, HTS, HnLTS [19], and CaGCN [5] are implemented within the SGRACE framework to guarantee consistent deployment and evaluation environments.

The proposed method, NHTS, consistently achieves the lowest ECE and NLL across datasets and bit-widths compared to other calibration techniques. While other GNN calibration approaches often rely on auxiliary GNN or attention mechanisms that significantly increase computational overhead, NHTS only uses lightweight entropy and structural feature computations and requires no modifications to the hardware design. Consequently, it does not consume additional FPGA resources such as LUTs, BRAM, or DSPs, and the overall latency remains determined by the SGRACE accelerator. As illustrated in Table II, on the Cora dataset with 4-bit quantization, NHTS reduces ECE to 2.81%, outperforming other calibration methods such as TS (9.82%), HTS (4.59%), and CaGCN (4.05%). Similarly, on other datasets, NHTS achieves the best results in both calibration and likelihood metrics, which allows it to outperform node-level and hierarchical baselines. The effectiveness of NHTS is more pronounced on larger datasets with stable neighborhood structures, where entropy-based calibration becomes more effective. For instance, on the Pubmed dataset with 2-bit quantization, NHTS achieves the lowest calibration error (ECE: 2.62%) and NLL (0.542) compared to other methods such as HTS and CaGCN. These results suggest that NHTS scales more effectively with dataset size.

Figure 3 illustrates the performance of the NHTS method comparing Top-1 error against ECE and NLL across all datasets and quantization bit-widths. In both plots, triangle markers (NHTS) consistently appear in the lower-left region, indicating both lower error and better calibration compared to the uncalibrated baseline (circular markers). This improvement is especially notable at the 4-bit and 2-bit quantization level, where effective calibration plays a crucial role.

VI. CONCLUSION

This paper presents NHTS, a novel approach that addresses the largely unexplored intersection of calibration and quantization in GNNs. While prior research has advanced these areas independently, this study uniquely offers a systematic analysis and solution for the calibration issues introduced by quantization in GNNs. NHTS computes tailored temperature scaling for each node by leveraging node-specific entropy, neighborhood uncertainty, and graph structural features, thereby enhancing confidence reliability without compromising accuracy. The method is implemented using the SGRACE hardware accelerator, enabling efficient low-bit training and inference on resource-constrained edge devices. Experimental results across multiple datasets and quantization levels confirm that NHTS achieves state-of-the-art calibration performance, significantly reducing both ECE and NLL. In future work, we aim at using NHTS to trigger hardware reconfiguration between different SGRACE precisions when the output confidence drops below a certain threshold. We also plan to apply NHTS to a broader range of datasets and evaluate its performance on more advanced models, such as Graph Attention Networks (GATs), in addition to the GCNs considered in this study, in order to better understand its generalization capability.

ACKNOWLEDGMENT

This research was funded by the Wallenberg AI autonomous systems and software (WASP) program funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

- [1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.
- [2] J. Nunez-Yanez, "Adaptive quantization of graph convolutional networks with hardware-aware on-device training," in *2024 IEEE Nordic Circuits and Systems Conference (NorCAS)*. IEEE, 2024, pp. 1–7.
- [3] G. Xia, S. Ha, T. Azevedo, and P. Maji, "An underexplored dilemma between confidence and calibration in quantized neural networks," *arXiv preprint arXiv:2111.08163*, 2021.
- [4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *International Conference on machine learning*. PMLR, 2017, pp. 1321–1330.
- [5] X. Wang, H. Liu, C. Shi, and C. Yang, "Be confident! towards trustworthy graph neural networks via confidence calibration," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 768–23 779, 2021.
- [6] H. H.-H. Hsu, Y. Shen, C. Tomani, and D. Cremers, "What makes graph neural networks miscalibrated?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 775–13 786, 2022.
- [7] J. Nunez-Yanez and H. M. Jeddi, "Sgrace: Scalable architecture for on-device inference and training of graph attention and convolutional networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.
- [8] S. A. Tailor, J. Fernandez-Marques, and N. D. Lane, "Degree-quant: Quantization-aware training for graph neural networks," *arXiv preprint arXiv:2008.05000*, 2020.
- [9] I. Colbert, A. Pappalardo, and J. Petri-Koenig, "A2q: Accumulator-aware quantization with guaranteed overflow avoidance," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 989–16 998.
- [10] M. Yan, L. Deng, X. Hu, L. Liang, Y. Feng, X. Ye, Z. Zhang, D. Fan, and Y. Xie, "Hygcn: A gcn accelerator with hybrid architecture," in *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2020, pp. 15–29.
- [11] T. Geng, A. Li, R. Shi, C. Wu, T. Wang, Y. Li, P. Haghi, A. Tumeo, S. Che, S. Reinhardt *et al.*, "Awb-gcn: A graph convolutional network accelerator with runtime workload rebalancing," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020, pp. 922–936.
- [12] J. Li, A. Louri, A. Karanth, and R. Bunescu, "Genax: A flexible and energy-efficient accelerator for graph convolutional neural networks," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2021, pp. 775–788.
- [13] J. Nunez-Yanez, "Accelerating graph neural networks in pytorch with hls and deep dataflows," in *International Symposium on Applied Reconfigurable Computing*. Springer, 2023, pp. 131–145.
- [14] W. Yuan, T. Tian, Q. Wu, and X. Jin, "Qegcn: An fpga-based accelerator for quantized gcns with edge-level parallelism," *Journal of Systems Architecture*, vol. 129, p. 102596, 2022.
- [15] A. S. Mozafari, H. S. Gomes, W. Leão, S. Janny, and C. Gagné, "Attended temperature scaling: a practical approach for calibrating deep neural networks," *arXiv preprint arXiv:1810.11586*, 2018.
- [16] J. Zhang, B. Kailkhura, and T. Y.-J. Han, "Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning," in *International conference on machine learning*. PMLR, 2020, pp. 11 117–11 128.
- [17] D. Zhuang, C. Jiang, Y. Zheng, S. Wang, and J. Zhao, "Gets: Ensemble temperature scaling for calibration in graph neural networks," *arXiv preprint arXiv:2410.09570*, 2024.
- [18] J. Kuang and A. Wong, "On calibration of modern quantized efficient neural networks," *arXiv preprint arXiv:2309.13866*, 2023.
- [19] S. A. Balanya, J. Maroñas, and D. Ramos, "Adaptive temperature scaling for robust calibration of deep neural networks," *Neural Computing and Applications*, vol. 36, no. 14, pp. 8073–8095, 2024.