

Study of Hardware Acceleration of Graph Neural Networks

Surya Rao Rayarao^{1*}

^{1*}*Department of Statistics and Data Sciences at The University of Texas at Austin, TX, United States*

ARTICLE INFO

Article history:

Received: 20240815

Received in revised form: 20240825

Accepted: 20240825

Available online: 20240914

Keywords:

Power estimation;

Machine learning;

Graph neural network;

Link Prediction;

Attributed Network;

Hashing.

ABSTRACT

The size and complexity of graphs increase, the computational demands of GNNs become a major bottleneck. To address this challenge, researchers have explored hardware acceleration techniques to speed up GNN computations. This paper presents an overview- of existing hardware acceleration methods for GNNs, including specialized hardware designs and optimizations. We discuss the advantages and limitations of these approaches and highlight the key factors to consider when designing hardware accelerators for GNNs. Furthermore, we present potential directions for future research in this domain, aiming to unlock the full potential of GNNs through efficient hardware acceleration.

Researchers are exploring various approaches for hardware acceleration of GNNs, including custom-designed hardware, accelerators like GPUs and TPUs, and software optimizations for existing hardware. As this research area progresses, it has the potential to revolutionize how graph-based data is processed, enabling more advanced and efficient solutions across a wide range of industries and domains.

The COPRAS method requires identifying selection criteria, evaluating information related to these criteria, and developing methods to evaluate Meeting the participant's needs Criteria for doing in order to assess the overall performance of the surrogate. Decision analysis involves a Decision Maker (DM) Situation to do consider a particular set of alternatives and select one among several alternatives, usually with conflicting criteria. For this reason, the developed complexity proportionality assessment (COPRAS) method can be used.

Hardware acceleration of GNNs. QM is got the first rank whereas the live journal is having the Lowest rank.

2024 Sciforce Publications. All rights reserved.

*Corresponding author. Tel.: +1 (913) 413-6752e-mail: suryarao.r@gmail.com

Introduction

Graph neural networks (GNNs) have expanded the capabilities of machine learning by effectively handling graph-structured inputs. These networks have demonstrated superior performance compared to existing methods across various tasks, spanning from predicting molecular properties to identifying communities within networks. However, the unique memory access and data manipulation demands of GNNs make them unsuitable for conventional execution platforms, including prevalent machine learning accelerators. To address this limitation, a novel architecture has been proposed. This architecture not only delivers the substantial computational throughput required by GNN models but also incorporates specialized hardware components designed to efficiently manage the complex data movement inherent in GNN calculations. Through empirical

evaluation, it has been proven that this architecture significantly surpasses existing execution platforms in terms of inference speed. For instance, it achieves a performance increase of 7.5 times over GPUs and an 18-fold improvement over CPUs, while maintaining equivalent bandwidth. [Reference: [1]] Network virtualization technology heavily relies on Virtual Network Embedding (VNE), a crucial aspect. Previous research predominantly focused on enhancing resource efficiency, often overlooking scalability as a core objective. Consequently, with the growing size and increasing demand, the effectiveness of these approaches diminishes.

Existing solutions aimed at tackling this challenge are either not applicable in multi-resource scenarios or fail to consider the

simultaneous optimization of physical servers and the network infrastructure. In this investigation, we introduce GraphViNE, a VNE solution designed with parallelizability in mind, employing spatial Graph Neural Networks (GNN). The key innovation is the server clustering approach employed by GraphViNE, which guides the embedding process to achieve faster runtimes and improved performance. Through extensive simulation-based assessments, we demonstrate that GraphViNE's parallelization significantly reduces the runtime by a factor of eight. Furthermore, when compared to other simulated algorithms, GraphViNE enhances the revenue-to-cost ratio by approximately 18%. [2]

A groundbreaking graph neural network (GNN) model enhanced by GPU acceleration has been developed for accurately and swiftly estimating vector-based average power. This novel approach involves representing a net list as a graph, incorporating register states and unit inputs from RTL simulation as features, and utilizing combinational gate toggle rates as labels. During the training process, the GRANITE model learns to predict average toggle rates through combinational logic. Subsequently, the trained GNN model can rapidly infer average toggle rates for new workloads or net lists within seconds. GRANITE achieves an impressive speedup of over 18.7 times, while maintaining a mere 5.5% margin of error compared to traditional power analysis methods that rely on gate-level simulations across various benchmark circuits. [3]

The real world is brimming with interconnected systems. One of the primary challenges when dealing with data organized in network structures is predicting links, which involves predicting whether a connection exists between two nodes. Traditional approaches rely on explicitly calculating similarity between concise representations of nodes, achieved by embedding each node into a lower-dimensional space. Hashing methods have been effectively employed to generate these node representations within the Hamming space, aiming to manage the resource-intensive similarity calculations in link prediction. However, the use of randomized hashing techniques or inefficiencies in learning-to-hash methods during the embedding process have led to reduced accuracy in hashing-based link prediction algorithms. We introduce a straightforward and successful model called #GNN that strikes a balance between precision and efficiency. By employing randomized hashing for message propagation and capturing higher-order relationships within the #GNN framework, this model rapidly generates node representations in the Hamming space, facilitating accurate link prediction. [4]

Sparse-Dense Matrix Multiplication (SpMM) is crucial for accelerating Graph Neural Networks (GNNs) and ensuring compatibility with various frameworks. However, modern SpMM techniques, which employ advanced sparse matrix representations, can introduce significant preprocessing overhead due to their compatibility considerations. Unlike Sparse Matrix-Vector (SpMV) optimizations, which don't seamlessly apply to SpMM, resulting in inefficient and scattered global memory access, the GE-SpMM1 approach employs the CSR format. This format aligns with GNN frameworks, facilitating integration without format modification overhead. To

enhance efficient memory access for both sparse and dense data in the global memory, coalesced row caching is employed. Furthermore, to minimize redundant data loading across GPU warps, a coarse-grained warp merging strategy is adopted. Empirical experiments on a real-world graph dataset demonstrate speedups of up to 1.41 compared to Nvidia cuSPARSE and up to 1.81 compared to GraphBLAST. The integration of GE-SpMM into GNN frameworks results in speedups of up to 3.67 for popular GNN models such as GCN and GraphSAGE. We create a versatile abstract framework that allows us to apply Graph Neural Network (GNN) models to predict traffic patterns. This involves treating a traffic scenario as a graph in which vehicles interact. GNNs offer computational efficiency, a large capacity for modeling, and a built-in ability to represent interactions among traffic participants.

We evaluate two advanced GNN architectures and make several adaptations to suit our specific context. Our results show that, compared to a model that disregards interactions, predictive accuracy improves by 30% in scenarios characterized by substantial interactions. This underscores the importance of accounting for interactions and illustrates the applicability of graph-based modeling. Consequently, GNNs prove to be a valuable enhancement to traffic prediction systems due to their capability in handling this aspect. Graph neural networks (GNNs) have emerged as a potent technique for processing non-euclidean data structures in various domains like social networks and e-commerce. However, their application to real-world systems with large and sparse graph data presents challenges due to the considerable computational and memory demands, which strain CPUs and GPUs in terms of energy and resources. To address this, we introduce the EnGN accelerator design, aiming to enable efficient and high-throughput processing of massive GNNs.

The EnGN design focuses on optimizing three crucial stages of GNN propagation, which are common computational patterns in GNNs. These stages are accelerated through the proposed EnGN. To handle the issues of poor locality in sparsely and randomly connected vertices, we introduce the ring-edge-reduce (RER) dataflow, along with the RER PE-array that implements this dataflow. This setup supports the necessary phases simultaneously. A graph tiling approach is employed to accommodate large graphs within EnGN. Comparatively, EnGN outperforms CPU, GPU, and the state-of-the-art GCN accelerator HyGCN in terms of performance speedup and energy efficiency. [7]

Convolutional neural networks (CNNs) have demonstrated their effectiveness in solving high-dimensional regression and classification problems within Euclidean domains. Recently, there has been a growing interest in geometric deep learning, also referred to as geometric generalization to non-Euclidean domains, due to its potential in pattern recognition and regression for graph-structured data. In this context, we propose an alternative orthonormal system called the Haar basis for graphs. We introduce the Haar convolution, a novel graph convolution technique tailored for Graph Neural Networks (GNNs). Leveraging the sparsity and localized nature of the

Haar basis on graph-structured data, we achieve efficient computation through fast Haar transforms (FHTs). This leads to a substantial enhancement in the computational efficiency of GNNs, as the Haar convolution ensures linear computational complexity. Our innovation culminates in the creation of HANet, a novel category of deep convolutional neural networks designed for graphs. Empirical evaluations on real graph datasets demonstrate HANet's exceptional performance and efficiency in classification and regression tasks. Notably, our method represents the first rapid algorithm for spectral graph convolution by carefully selecting an orthogonal basis on the graph—an essential step in developing spectral-based GNN models. In summary, the paper's principal contributions can be categorized into three key areas. [8]

This upgraded iCGCNN model showcases its effectiveness through two distinct examples. Firstly, when trained and validated on a dataset of 180,000 and 20,000 thermodynamic stability entries respectively, derived from density functional theory (DFT) calculations in the Open Quantum Materials Database (OQMD), iCGCNN achieves a significantly enhanced predictive accuracy—20% higher than the original CGCNN. This improvement is further validated on a separate test set containing 230,000 entries. Secondly, iCGCNN's capability is evident in its achievement of a success rate of 31% during a high-throughput search for materials possessing the ThCr₂Si₂ structure-type. This success rate surpasses an undirected high-throughput search by a substantial factor of 155 and also outperforms the original CGCNN by 2.4 times. In the pursuit of discovering novel materials, we employed both CGCNN and iCGCNN to conduct 757 density functional theory (DFT) computations on 132,600 compounds for elemental decorating of the ThCr₂Si₂ prototype crystal structure. This approach significantly increased the computational efficiency of the high-throughput search by a factor of 65. These findings underscore the potential of iCGCNN to expedite the identification of crystalline compounds with noteworthy attributes, thereby accelerating the high-throughput discovery of novel materials. [8]

Optimal controllers have been developed for a diverse array of issues, spanning from restricted consensus in multiagent systems to load control in electrical grids and throughput management in wireless networks. However, these controllers are centralized solutions, necessitating access to the entire system's real-time status. While centralized controllers are conceptually ideal, their practical implementation and scalability face limitations. In contrast, decentralized controller design hinges on the communication network formed by the constituent agents within the system. These agents are restricted to exchanging information solely with proximate agents. This distributed information framework serves as the basis for formulating a decentralized controller. [9]

Convolutional neural networks (CNNs) serve as a notable illustration of how effectively leveraging data structures within temporal sequences and images has transformed the landscape of machine learning in the past decade. CNNs employ temporal or spatial convolutions to adapt to extensive scenarios, acquire

adept nonlinear mappings, and mitigate the risk of overfitting. Moreover, CNNs offer a degree of mathematical tractability, enabling the derivation of theoretical performance boundaries concerning domain perturbations. However, CNNs prove inefficient for learning from irregular network data due to their confinement to convolutions applicable solely to data residing in regular domains. [10]

Deep neural networks (DNNs) have made remarkable advancements across various domains like speech and image recognition, as well as natural language processing. This progress has enabled their application in practical scenarios such as self-driving cars, search engines, recommendation systems, and more. Convolutional neural networks (CNNs) have particularly excelled in computer vision tasks. In the realm of graph-structured data like social networks and knowledge graphs, researchers have introduced graph convolutional networks (GCNs) as a means to apply convolutional techniques. In GCNs, a single convolution operation works to aggregate and modify feature information from a node's immediate graph connections. By stacking multiple such convolutions, a node's information spreads extensively across the graph, effectively leveraging both feature details and graph structure. GCNs have shown impressive model accuracy in various real-world applications. For instance, in recommendation systems, they can learn features from the user-item graph to generate higher-quality recommendations. In machine learning applications, the prevalence of large graph datasets, containing intricate relationships among potentially billions of elements, has increased. To effectively handle the complexities of these graphs, Graph Neural Networks (GNNs) have gained prominence. [14]

This multiscale graph adapts dynamically across network layers during training. To achieve this, a multiscale graph computational unit (MGCU) is introduced, enabling the extraction of characteristics across various scales and the fusion of features between these scales. The architecture follows an encoder-decoder structure and remains agnostic to specific action categories. The encoder employs a series of MGCUs to grasp motion features, while the decoder generates future poses using a graph-based gate recurrent unit. [15]

Materials and Method

2.1 Graphs: Graphs are visual representations of data that use a set of points (vertices or nodes) connected by lines or curves (edges) to show the relationship between various elements. They provide a clear and concise way to illustrate patterns, trends, and correlations within the data, making complex information easier to understand and analyze. The main purpose of graphs is to present data in a more visually appealing and insightful manner. By plotting data points on a graph, it becomes easier to identify patterns, trends, outliers, and relationships that might not be immediately apparent when looking at raw numbers or textual descriptions. Graphs are widely used in various fields, including mathematics, science, economics, social sciences, and more, to present and interpret data effectively

2.2 Total nodes: The total number of nodes is an important metric when analyzing the performance of tree-based algorithms and when assessing the efficiency of operations on the tree data structure. It is used to measure the size and complexity of the tree, which can impact the time and space complexity of various operations, such as searching, insertion, deletion, and traversal.

2.3 Total edges: In graph theory, the term "total edges" typically refers to the total number of edges in a graph. A graph is a mathematical representation of a set of objects (vertices or nodes) connected by links (edges). Each edge in the graph represents a relationship or connection between two vertices. The total number of edges in a graph can vary depending on the specific graph structure and the number of vertices it contains. For an undirected graph (where edges have no direction), the total number of edges is often denoted by "E," and it represents the count of all connections between pairs of vertices. In a directed graph (where edges have direction), each edge connects a specific starting vertex to an ending vertex, and the total number of edges is again represented by "E." For example, in a simple undirected graph with four vertices labeled A, B, C, and D, the total number of edges might be 3, and these edges could be represented as {AB, BC, CD}, indicating the connections between the vertices. In the case of a directed graph, the edges would have an arrow to indicate the direction, like {A → B, B → C, C → D}.

2.4 Vertex features: In the context of graphs and network analysis, "vertex features" refer to attributes or characteristics associated with each individual node (also known as vertices) in the graph. Nodes are the fundamental building blocks of a graph, and vertex features provide additional information about these nodes, helping to describe and differentiate them. For example, in a social network graph, each node could represent a person, and the vertex features might include attributes such as age, gender, location, occupation, and interests. In a transportation network graph, the nodes might represent cities or intersections, and the vertex features could include population, traffic density, or road conditions. Vertex features play a crucial role in various applications of graph analysis by considering the features of each vertex, algorithms can make more informed decisions and better understand the structure and patterns within the graph.

2.5 Cora: The Cora dataset consists of academic publications, where each publication is represented as a node in a citation network. When researchers discuss hardware acceleration of GNNs with the "CORA" dataset, they are likely referring to the application of specialized hardware, TPUs, to speed up the computation and training of GNN models on this particular dataset. This is because GNNs can be computationally intensive, especially when dealing with large graphs, and leveraging hardware acceleration can significantly reduce the training time and improve the efficiency of these models.

2.6 Cite seer: The paper or research related to this topic may discuss various hardware implementation strategies, such as using specialized hardware architectures like GPUs, TPUs, or FPGA to accelerate GNN computations. Hardware acceleration aims to improve the efficiency and performance of GNNs,

enabling faster and more scalable solutions for graph-related tasks.

Method: In 1996 in Lithuania COPRAS (Complex Proportion evaluation) method was developed Construction, economics, real estate and management. One of the articles assesses the risks involved in construction projects. The assessment is based on various multi-objective assessment methods. The risk assessment indices are selected considering the interests, objectives and factors of the countries that influence the construction efficiency and real estate price increase [16] to describe and consider the task model. Complex Proportionality Assessment (COPRAS) Method Similar to any Many other criteria will make the decision (MCDM) tool, first Proposed COBRAS method of several related criteria Basically for alternatives Used to prioritize criterion weights. This method is better and Worst-Best Solutions Best decision considering Selecting alternatives [17].

Cobras approach is used for device tool choice; Because of this the triangle Ambiguous numbers are selected their computational performance. Three area specialists are selected to assign weights and by way of combining the fuzzy cobra's method, System 1 (MC1) and device 2(MC2) similarly are ranked, with way of machine three and four. -based totally approach is utilized in mixture with fuzzy. COPRAS assess the complexity of consumer dating management (CRM) performance. A combined choice matrix is obtained from a panel of 20 specialists offered 3 options with set, and 5 criteria Assessments are done [18].

COPRAS to resolve MCDM issues, wherein the weights of the criteria and Performance ratings of alternatives are absolute Based on linguistic terms are calculated. Comparison of criteria Importance calculated and Cobra's method become used to assess renovation strategies [19].

This have a look at ambitions to develop the impact of latest overall performance metrics in TPM and COPRAS in an ambiguous context Primarily multi-criteria selection based on opinions Use the do method. COPRAS method changed into The most relevant social media platform Rank and choose is used. Proposed Applicability of the structure We proved and proved the character [22].

COPRAS (Complex Proportionality Assessment) To examine Cumulative of an alternative Performance, it is essential become aware of the maximum vital criteria, examine the options and compare the facts Depending on those criteria to fulfil the wishes of the DMs to compare gradesevaluation involves a situation in which a DM must pick amongst several downloaded alternatives given a selected set of commonly conflicting standards. For this motive, the developed complex proportionality evaluation (COPRAS) method can be used in real situations, alternatives The criteria for assessment are vague is related to the factor, And the values of the standards are real Cannot be expressed with numbers [23].

3. RESULTS AND DISCUSSION

TABLE 1. Hardware Acceleration of Graph Neural Networks

	Graphs	Total nodes	Total edges	Vertex features
Cora	1	2710	54232	1438
cite seer	1	3330	4745	3710
pub med	1	19712	44335	510
QM	1000	12320	12095	16
DBLP	1	2660	2664	1

This table provides a comparison of various graph datasets based on the number of graphs, total nodes, total edges, and the number of vertex features they contain. The datasets include Cora, Cutesier, Pub Med, QM, and DBLP.

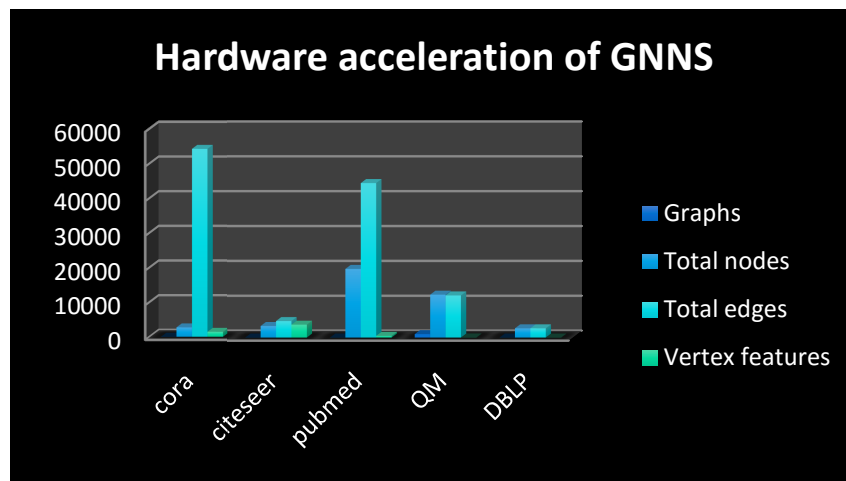


Figure 1. Hardware Acceleration of Graph Neural Networks

Table 2. Normalized Data

Normalized Data			
Graphs	Total nodes	Total edges	Vertex features
0.0010	0.0665	0.4593	0.2534
0.0010	0.0818	0.0402	0.6537
0.0010	0.4839	0.3755	0.0899
0.9960	0.3025	0.1024	0.0028
0.0010	0.0653	0.0226	0.0002

Table 2 shows hardware acceleration of GNNS. Normalized Data for graphs total nodes, total edges, vertex features Normalized value.

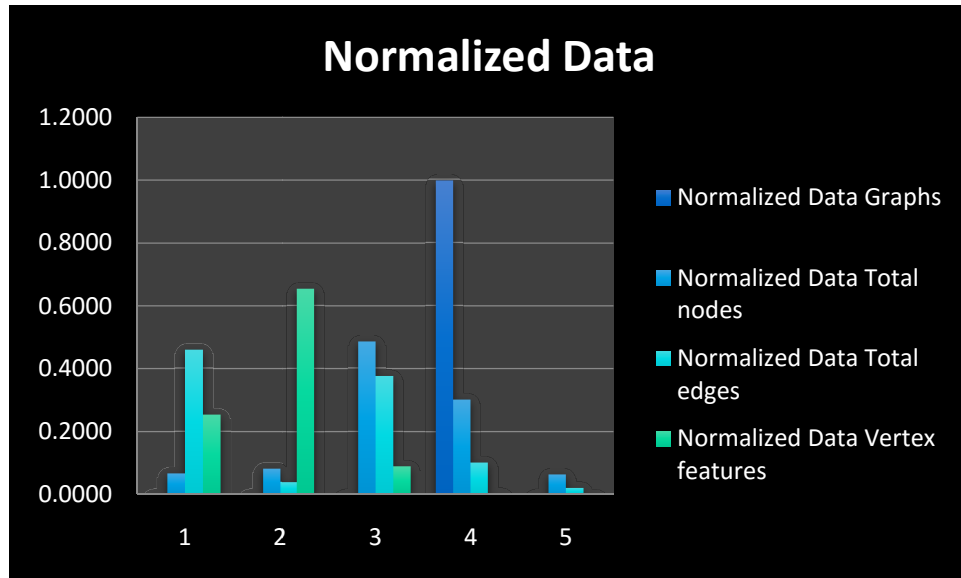


Figure 2 shows hardware acceleration of GNNS. Normalized Data for graphs, total nodes, total edges , vertex features.Normalized value.

TABLE 3. Weight ages

Weight			
0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25
0.25	0.25	0.25	0.25

Table 3 shows Weight ages used for the analysis. We take same weights for all the parameters of the analysis

TABLE 4. Weighted normalized decision matrix

Weighted normalized decision matrix			
0.00	0.02	0.11	0.06
0.00	0.02	0.01	0.16
0.00	0.12	0.09	0.02
0.25	0.08	0.03	0.00
0.00	0.02	0.01	0.00

Table 4 shows the weighted normalized decision matrix for graphs, total nodes, total edges, vertex features is also Multiple value.

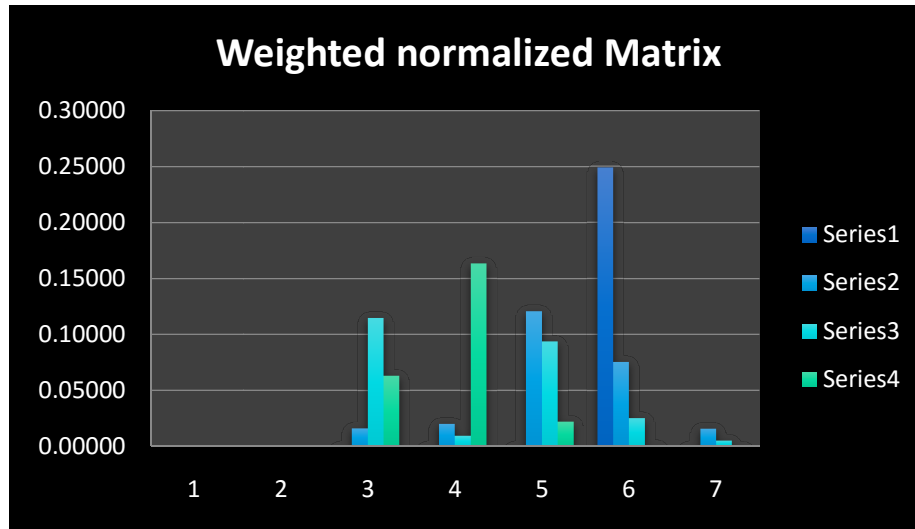


Figure 3 shows the weighted normalized decision matrix for graphs, total nodes, total edges, vertex features is also Multiple value.

Table 5. Bi, Ci, Min(Ci)/CiN

Bi	Ci	
0.017	0.178	
0.021	0.173	
0.121	0.116	
0.325	0.026	
0.017	0.006	

Table 5 shows the hardware acceleration of GNNs Bi, Ci, Min (Ci)/Ci Graphs, total nodes, total edges vertex features. it is sum of minimum value.

Table 6. Final Result of hardware acceleration of GNNs

Min(Ci)/Ci	Qi	Ui	
0.0319	0.029	7.1158	
0.0328	0.033	8.1334	
0.0489	0.140	34.3986	
0.2160	0.406	100.0000	
1.0000	0.393	96.7418	

Table 6 shows the final result of COPRAS for hardware acceleration of GNNs

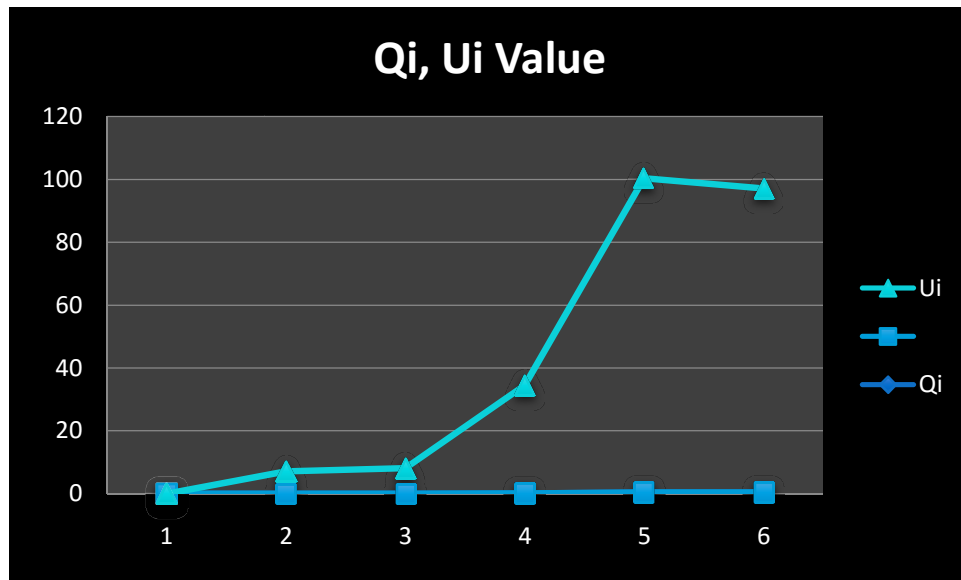


Figure 4 Qi, Ui Value

Figure 4 shows the final result of COPRAS for hardware acceleration of GNNs

Table 7. Ranks

	Rank
cora	5
citeseer	4
pubmed	3
QM	1
DBLP	2

Table 7 shows the ranks

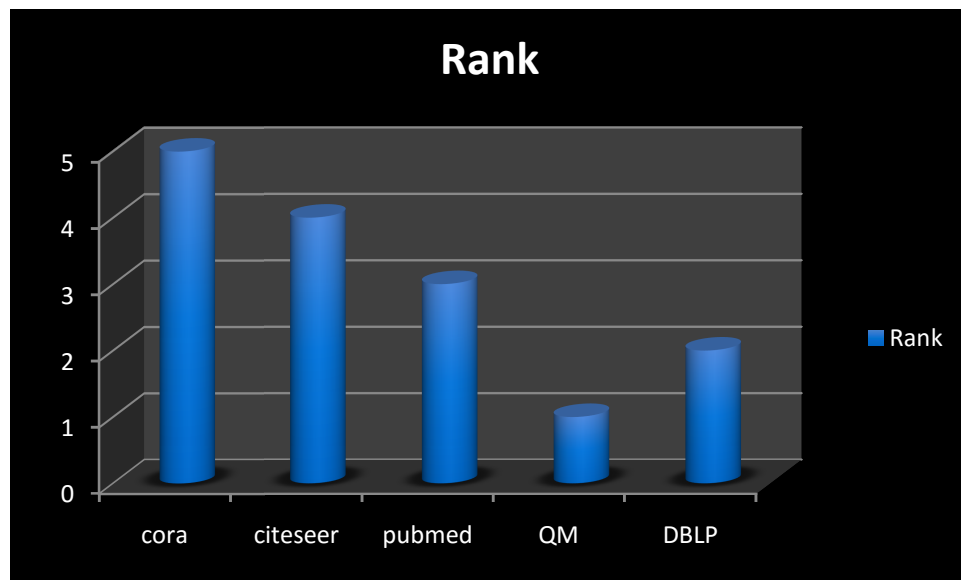


Figure 5 Shows Ranking of hardware acceleration of GNNs. QM is got the first rank whereas is the live journal is having the Lowest rank.

Conclusion

In conclusion, the hardware acceleration of Graph Neural Networks (GNNs) marks a significant advancement in the field of machine learning and graph analytics. Through the utilization of specialized hardware such as Graph Processing Units (GPUs), Field-Programmable Gate Arrays (FPGAs), and even custom-designed Application-Specific Integrated Circuits (ASICs), the performance and efficiency of GNN computations have been greatly enhanced. This hardware acceleration addresses the inherent challenges of GNN computations, which are characterized by their heavy reliance on graph structures and complex neighborhood aggregations. By leveraging parallelism, optimized memory access, and tailored architectures, GNNs can now be executed with remarkable speedup and energy efficiency, enabling the analysis of larger and more intricate graphs in real time. Furthermore, the synergy between software algorithms and hardware architectures has played a pivotal role in achieving these advancements. Researchers and engineers have collaborated to develop specialized GNN algorithms that are compatible with the strengths of various hardware platforms. This alignment between software and hardware has led to breakthroughs in both performance and versatility. Nonetheless, there remain challenges to address in this domain. The diversity of graph structures, the evolving landscape of hardware technologies, and the demand for adaptable solutions call for ongoing research and innovation. Additionally, the integration of hardware acceleration into existing machine learning frameworks and pipelines requires careful consideration to ensure seamless usability and maintainability. In summary, the hardware acceleration of Graph Neural Networks holds great promise for revolutionizing the analysis of graph data in various domains, from social networks to molecular chemistry. The collaborative efforts of researchers and engineers in refining hardware architectures and developing optimized algorithms are paving the way for more efficient, scalable, and powerful graph analytics systems. As this field continues to progress, we anticipate even more remarkable developments at the intersection of hardware and machine learning.

REFERENCES

1. Habibi, Farzad, Mahdi Dolati, Ahmad Khonsari, and Majid Ghaderi. "Accelerating virtual network embedding with graph neural networks." In *2020 16th International Conference on Network and Service Management (CNSM)*, pp. 1-9. IEEE, 2020.
2. Zhang, Yanqing, Haoxing Ren, and Bruce Khailany. "GRANNITE: Graph neural network inference for transferable power estimation." In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1-6. IEEE, 2020.
3. Wu, Wei, Bin Li, Chuan Luo, and Wolfgang Nejdl. "Hashing-accelerated graph neural networks for link prediction." In *Proceedings of the Web Conference 2021*, pp. 2910-2920. 2021.
4. Huang, Guyue, Guohao Dai, Yu Wang, and Huazhong Yang. "Ge-spm: General-purpose sparse matrix-matrix multiplication on gpus for graph neural networks." In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1-12. IEEE, 2020.
5. Diehl, Frederik, Thomas Brunner, Michael Truong Le, and Alois Knoll. "Graph neural networks for modelling traffic participant interaction." In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 695-701. IEEE, 2019.
6. Liang, Shengwen, Ying Wang, Cheng Liu, Lei He, L. I. Huawei, Dawen Xu, and Xiaowei Li. "Engn: A high-throughput and energy-efficient accelerator for large graph neural networks." *IEEE Transactions on Computers* 70, no. 9 (2020): 1511-1525.
7. Li, Ming, Zheng Ma, Yu Guang Wang, and Xiaosheng Zhuang. "Fast Haar transforms for graph neural networks." *Neural Networks* 128 (2020): 188-198.
8. Park, Cheol Woo, and Chris Wolverton. "Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery." *Physical Review Materials* 4, no. 6 (2020): 063801.
9. Gama, Fernando, Ekaterina Tolstaya, and Alejandro Ribeiro. "Graph neural networks for decentralized controllers." In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5260-5264. IEEE, 2021.
10. Ballamudi, S. "Interleaved Feature Extraction Model Bridging Multiple Techniques for Enhanced Object Identification" *Journal of Artificial Intelligence and Machine Learning*, 2023, vol. 1, no. 2, pp. 1-7. doi: <https://doi.org/10.55124/jbid.v1i2.253>
11. Gama, Fernando, Elvin Isufi, Geert Leus, and Alejandro Ribeiro. "Graphs, convolutions, and neural networks: From graph filters to graph neural networks." *IEEE Signal Processing Magazine* 37, no. 6 (2020): 128-138.
12. Tian, Chao, Lingxiao Ma, Zhi Yang, and Yafei Dai. "Pcgc: Partition-centric processing for accelerating graph convolutional network." In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 936-945. IEEE, 2020.
13. Zhu, Rong, 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).
14. Li, Maosen, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. "Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 214-223. 2020.
15. Yazdani, Morteza, Ali Alidoosti, and Edmundas Kazimieras Zavadskas. "Risk analysis of critical infrastructures using fuzzy COPRAS." *Economic research-Ekonomska istraživanja* 24, no. 4 (2011): 27-40. <https://doi.org/10.1080/1331677X.2011.11517478>
16. Aghdaie, Mohammad Hasan, Sarfaraz Hashemkhani Zolfani, and Edmundas Kazimieras Zavadskas. "Market segment evaluation and selection based on application of fuzzy AHP and COPRAS-G methods." *Journal of Business Economics*

- and Management 14, no. 1 (2013): 213-233. <https://doi.org/10.3846/16111699.2012.721392>
17. Kildienė, Simona, Arturas Kaklauskas, and EdmundasKazimierasZavadskas. "COPRAS based comparative analysis of the European country management capabilities within the construction sector in the time of crisis." Journal of Business Economics and Management 12, no. 2 (2011): 417-434.
18. Das, Manik Chandra, Bijan Sarkar, and Siddhartha Ray. "A framework to measure relative performance of Indian technical institutions using integrated fuzzy AHP and COPRAS methodology." Socio-Economic Planning Sciences 46, no. 3 (2012): 230-241. <https://doi.org/10.1016/j.seps.2011.12.001>
19. Dhiman, Harsh S., and Dipankar Deb. "Fuzzy TOPSIS and fuzzy COPRAS based multi-criteria decision making for hybrid wind farms." Energy 202 (2020): 117755. <https://doi.org/10.1016/j.energy.2020.117755>
20. Fouladgar, Mohammad Majid, Abdolreza Yazdani-Chamzini, Ali Lashgari, EdmundasKazimierasZavadskas, and ZenonasTurskis. "Maintenance strategy selection using AHP and COPRAS under fuzzy environment." International journal of strategic property management 16, no. 1 (2012): 85-104. <https://doi.org/10.3846/1648715X.2012.666657>
21. TuranogluBekar, Ebru, Mehmet Cakmakci, and Cengiz Kahraman. "Fuzzy COPRAS method for performance measurement in total productive maintenance: a comparative analysis." Journal of Business Economics and Management 17, no. 5 (2016): 663-684. <https://doi.org/10.3846/16111699.2016.1202314>
22. Zolfani, Sarfaraz Hashemkhani, Nahid Rezaeiniya, Mohammad Hasan Aghdaie, and EdmundasKazimierasZavadskas. "Quality control manager selection based on AHP-COPRAS-G methods: a case in Iran." Economic research-Ekonomskaitraživanja 25, no. 1 (2012): 72-86. <https://doi.org/10.1080/1331677X.2012.11517495>
23. Tavana, Madjid, Ehsan Momeni, Nahid Rezaeiniya, Seyed Mostafa Mirhedayatian, and HamidrezaRezaeiniya. "A novel hybrid social media platform selection model using fuzzy ANP and COPRAS-G." Expert Systems with Applications 40, no. 14 (2013): 5694-5702. <https://doi.org/10.1016/j.eswa.2013.05.015>
24. Kouchaksaraei, RamtinHaghnazar, Sarfaraz HashemkhaniZolfani, and Mahmood Golabchi. "Glasshouse locating based on SWARA-COPRAS approach." International Journal of Strategic Property Management 19, no. 2 (2015): 111-122. <https://doi.org/10.3846/1648715X.2015.1004565>