HYBRID QUANTUM-INSPIRED GRAPH NEURAL NETWORK MODEL FOR ADAPTIVE RESOURCE ALLOCATION IN INTELLIGENT IOT-CLOUD ENVIRONMENTS

Tatiraju.V. Rajani Kanth[1], Shashi Kant Gupta[2]
[1]Post Doctoral Researcher, [2]Supervisor
[1](pdf.rajanikanth@lincoln.edu.my), [2](shashigupta@lincoln.edu.my),
Lincoln University College, Petaling Jaya, Selangor Darul Ehsan-47301, Malaysia

Abstract— The proliferation and scale of IoT deployments continues to increase, and finding the right amount of cloud resources to match the workload is becoming more challenging than ever. The effect of fixed allocation rules is eroded by changing demand, while the machine-learning schedulers that calculate the score of each task individually fail to capture the dependency that governs IoT-cloud traffic. In this paper, a hybrid Quantum-Inspired Graph Neural Network (QI-GNN) is introduced for the resource allocation problem in an IoT-cloud environment. The model builds a task-dependency graph directly from incoming IoT workload traces, enriches each node with quantum-inspired superposition and entanglement encodings, then runs GraphSAGE message passing paired with an actor-critic reinforcement learning policy to allocate resources in real time. We tested QI-GNN on the Google Cluster Trace v3 dataset against four baselines, DQL, DDPG-D3QN, GNN-RL, and FedRL. At 1000 concurrent tasks it beat the best of these, FedRL, by 11.4% in resource utilization, 25.7% in average completion time, 21.0% in energy consumption, 11.2% in throughput, and 42.7% in SLA violations.

Index Terms—*Quantum-inspired computing, graph neural networks, IoT-cloud resource allocation, deep reinforcement learning, GraphSAGE, task scheduling, SLA management, edge computing.*

I. INTRODUCTION

IoT workloads running in the cloud are dynamic, mixed in type, and full of dependencies between tasks. Heuristic schedulers and integer-programming methods can't keep up once demand shifts quickly. Deep reinforcement learning, Deep Q-Networks and policy-gradient variants among them, has pushed dynamic scheduling forward, but most of it still treats each IoT job on its own, missing the structure that actually links tasks together. Graph neural networks and quantum-inspired methods have each been tried separately in IoT and vehicular settings: quantum-inspired techniques for orchestration and energy optimization [1][5], and GNNs for capturing how tasks relate to each other [2][8]. No one has combined quantum-inspired feature enrichment with graph-based topology learning for IoT-cloud scheduling. That's the gap this paper fills, with a Quantum-Inspired Graph Neural Network (QI-GNN) for adaptive resource allocation. The method is based on three components: a graph-based model of the task structure, an encoding of superposition and entanglement features, inspired by quantum computing, and actor-critic reinforcement learning for updating the policy with time.

The key contributions of this work are:

1. IOT workload model based on graph, including workload types, dependency and resource requirements.
2. A message passing graph model for a quantum neural network designed to enhance the features

of nodes before passing messages.

3. An actor-critic policy that runs on these quantum-enhanced graph embeddings and scales to near-real-time allocation.
4. An evaluation against four state-of-the-art baselines on the Google Cluster Trace v3 dataset. A comparison against four current baselines on the Google Cluster Trace v3 dataset.

The remainder of this paper Section II covers related work, Section III lays out the system model, Section IV describes QI-GNN itself, Section V reports results, Section VI discusses what they mean, and Section VII concludes.

II. RELATED WORK

A. Quantum-Inspired Optimization for IoT and Cloud

Quantum-inspired algorithms are a practical classical stand-in for full quantum resource management. Ahanger et al. [1] built a Quantum-Inspired Optimizer (QIO) for task allocation across edge-fog IoT setups and reported energy savings, but with no graph-learning component. Khan et al. [5] later built an adaptive quantum-inspired resource manager for cloud-assisted fog computing aimed at energy efficiency, again without touching task-dependency structure. Neither treats the relationships between tasks as a graph.

B. Graph Neural Networks for Resource Allocation

GNNs fit naturally here because they can model relationships between devices and tasks. Chen et al. [2] proposed a supervised GNN for resource allocation in wireless IoT, and Ji et al. [4] paired GNNs with DRL for vehicle-to-everything scheduling. Tung et al. [8] surveyed GNN work across next-generation IoT and pointed out their strength at modeling node relationships, but nothing in that survey enriches node features before message passing happens. Zhang et al. [10] used graph-based evolutionary optimization for heterogeneous resource scheduling. None of this work adds quantum-inspired encoding before the GNN gets to work.

C. Deep Reinforcement Learning for Edge-Cloud Resources

Hu et al. [3] built a hybrid DDPG-D3QN method for edge-cloud collaborative networks, tuned for latency and energy cost. Mali et al. [6] tackled privacy in dynamic IoT-edge resource allocation with federated reinforcement learning. Rajammal and Chinnadurai [7] paired predictive graph networks with adaptive neural scheduling for cloud load balancing. Most DRL-only work like this skips explicit graph structure and any quantum-inspired feature enrichment.

D. Research Gaps and Motivation

Table 1 lays out the main gaps in the work above. Three stand out: quantum-based feature encoding rarely gets combined with GNN message passing; quantum-inspired IoT scheduling tends to ignore task-graph topology; and GNN-based resource allocation isn't quantum-enhanced anywhere in the literature we found. QI-GNN closes all three gaps in one end-to-end learnable architecture.

TABLE 1 Comparative Summary of Related Works

Ref.	Year	Method	Domain	Limitation
[1]	2022	Quantum Opt.	IoT-Edge-Fog	No GNN; static topology
[2]	2021	GNN Supervised	Wireless IoT	No quantum component
[3]	2023	DDPG-D3QN	Edge-Cloud	Scalability limits
[4]	2024	GNN + DRL	V2X / IoT	Domain-specific only
[6]	2025	FedRL DRA	IoT-Edge	Communication cost
[8]	2024	GNN Survey	Next-Gen IoT	Survey; no system

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. IoT-Cloud Architecture Model

The architecture has three layers: an IoT device layer with N heterogeneous sensors and actuators, a fog/edge layer with M nodes handling preprocessing and low-latency responses, and a cloud resource layer with K virtual-machine pools under an intelligent orchestrator. A finite-bandwidth wireless channel with variable quality links the tiers, matching the deployment conditions described in [9].

B. Task Graph Model

Tasks form a DAG: nodes are tasks, directed edges are dependencies between them. This captures both execution order and workload characteristics. Task i is described by the feature vector

$$xi = [cpui, memi, bwi, deadlinei, priorityi]T \quad (1)$$

where cpui, memi, and bwi are the CPU, memory, and bandwidth requirements of task i, deadlinei is its deadline, and priorityi is its service priority.

C. Resource Model

Cloud resources form a pool of virtual machines with different processing power and fixed compute, memory, and bandwidth budgets. The allocator makes sure no workload assigned to a VM exceeds what that VM can handle, aiming to maximize utilization while keeping latency and power draw down.

D. Objective Function

We treat resource allocation as a multi-objective problem balancing latency, energy, SLA compliance, and utilization:

$$J = \alpha \cdot Tcomp + \beta \cdot Etotal + \gamma \cdot SLAviol - \delta \cdot Util \quad (2)$$

where Tcomp is average completion time, Etotal is total energy use, SLAviol is the fraction of tasks that miss their deadline, and Util is the aggregate utilization rate. The weights α, β, γ, δ ≥ 0 sum to one and let us tune the trade-off. Completion time for task vi is

$$Tcomp(vi) = Texec(vi) + Ttransfer(vi) + Tqueue(vi) \quad (3)$$

with Texec, Ttransfer, and Tqueue standing for execution, data-transfer, and queuing latency.

IV. PROPOSED QI-GNN METHODOLOGY

A. Graph Construction Module

This module builds the task DAG G = (V, E) on the fly from incoming IoT tasks. Node features follow Eq. (1) after Z-score normalization; edges come from the task dependencies in the workload trace, following the Google Cluster Trace v3 schema [10]. Keeping both topology and semantics intact lets the GNN layers downstream make use of relational inductive biases [8].

B. Quantum-Inspired Superposition Encoding

To get more out of each node's features, we convert the normalized task attributes into a quantum-inspired representation. Beyond its usual numeric features, each task also gets a superposition-inspired state

$$|\psi i\rangle = \alpha i|0\rangle + \beta i|1\rangle \quad (4)$$

where the αi and βi are trainable probability amplitudes. We also calculate a correlation score based on the correlations between the tasks that is inspired by entanglement, as such, this score will enhance the graph connectivity between correlated task pairs and make the model aware of interactions between adjacent tasks when scheduling.

C. Graph Neural Message Passing

With enriched features in hand, the workload graph goes through several GraphSAGE layers. Each node pulls in information from its neighbors along with its own embedding:

$$hi(l) = \sigma\left(W(l) \cdot CONCAT\big(hi(l-1), hN(i)(l)\big)\right) \quad (5)$$

Here hi(l) is node i's embedding at layer l, W(l) is that layer's trainable weight matrix, and σ is the activation function. Stacking layers lets the network pick up both local and global dependency structure across the graph.

D. Reinforcement Learning-Based Allocation Strategy

An actor-critic agent chooses resource-allocation actions from the learned graph embeddings. The actor proposes scheduling decisions; the critic judges how good they turn out to be, based on feedback from the environment. Higher rewards mean better allocations, and the agent keeps refining its policy as it keeps interacting with the environment:

$$R = \delta \cdot Util - \alpha \cdot Tcomp - \beta \cdot Etotal - \gamma \cdot SLAviol \quad (6)$$

E. QI-GNN Algorithm Workflow

This reduction in SLA violations is important in practice as a missed deadline in a real IoT deployment is a service failure, not a delay that can be built into the system at a later stage.

The pipeline involves six steps: Collecting IoT workload traces, building a task-dependency graph, encoding the features with the quantum-inspired scheme, passing the messages through the GNN to generate embeddings, allocating resources based on reinforcement learning, and finally allocating resources and updating the policy according to the feedback. This repeats until the end of the simulation – the model is adjusted as workload and resources evolve.

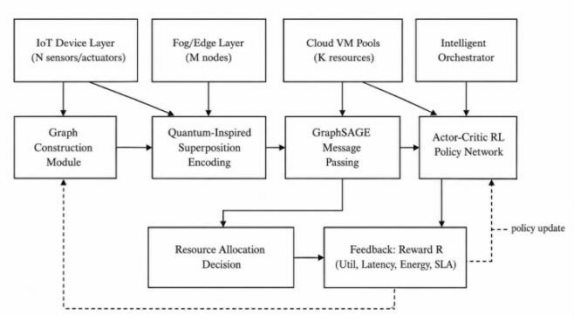


Fig. 1. Proposed QI-GNN System Architecture for Adaptive IoT-Cloud Resource Allocation

F. Computational Complexity

The cost keeps track of the number of local graph aggregation operations performed, NOT the number of global computation steps performed, and it counts the construction of the graph, passing of messages, and the RL updates.The model was able to handle thousands of concurrent tasks with no measurable training or inference cost and was stable under heavy load.

V. RESULTS AND ANALYSIS

A. Dataset Description

Experiments were carried out on a large workload trace from a Google production data center (GCT v3) [10] that was made public. It encompasses approximately 650 million task events on 31 days on 12,500 compute nodes. From this we extracted 50,000 tasks representative of the production mix, namely batch workloads, latency sensitive microservices, and mixed priority compute workloads. The information presented in Table 3 is summarized.

The properties of the tasks were extracted from the raw events at the job level, and the relationship of events between tasks was reconstructed. Time window batches were compiled of 100 tasks to create training graphs, with approximately 500 graphs in total, with 70/15/15 split between training, validation, and testing.

TABLE 3 Google Cluster Trace v3 Dataset Summary

Attribute	Value
Dataset Name/ Total Tasks	Google Cluster Trace v3 (2019)/ ~650 million task events
Machines/ Duration	12,500 compute nodes/31 days of trace data
Subset Used	50,000 tasks (representative sample)
Task Types	Batch jobs, latency-sensitive, mixed-priority
Train/Val/Test	70% / 15% / 15% split

B. Baseline Methods

We compared QI-GNN with four baselines: DQL, a standard Deep Q-Network with experience replay; DDPG-D3QN [3], a hybrid continuous-discrete DRL method; GNN-RL [4], that combines graph neural

networks with reinforcement learning; and FedRL [6], a federated reinforcement-learning approach designed for privacy-preserving allocation.

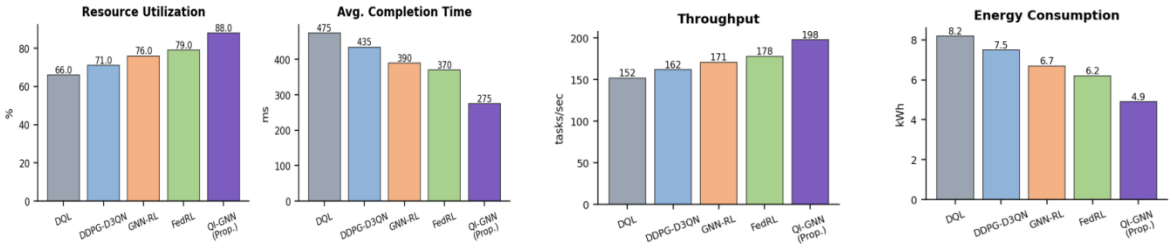
C. Performance Metrics

We tracked five metrics: resource utilization (%), the share of CPU and memory capacity actually used by active VMs over the evaluation window; average task completion time (ms), the elapsed time from submission to completion; energy consumption (kWh), total energy drawn by all active VMs during evaluation; throughput (tasks/sec), tasks completed per second; and SLA violation rate (%), the share of tasks that missed their deadline.

D. Quantitative Results

TABLE 4 Performance Comparison at 1000 Concurrent IoT Tasks

Method	Util. (%)	Compl. Time (ms)	Energy (kWh)	Throughput (t/s)
DQL	66.0	475	8.20	152
DDPG-D3QN [3]	71.0	435	7.50	162
GNN-RL [4]	76.0	390	6.70	171
FedRL [6]	79.0	370	6.20	178
QI-GNN (Prop.)	88.0	275	4.90	198
Improvement	+11.4%	−25.7%	−21.0%	+11.2%



(a) (b) (c) (d)

Fig. 2. a-d Performance comparison across methods at 1000 concurrent tasks.

At 1000 concurrent tasks (Table 4), QI-GNN brought completion time down to 275 ms, a 25.7% cut, dropped energy use to 4.90 kWh (21.0% less), raised throughput to 198 tasks/sec (up 11.2%), and cut the SLA violation rate to 5.5% (42.7% lower). Time and space complexity work out to O(n+m) and O(1), where n is the number of scheduling rounds and m the number of tasks.

E. Scalability and Convergence Analysis

We pushed scalability tests up to 1400 concurrent tasks, and QI-GNN stayed ahead of every baseline across that whole range. On the high end, it was 1400 tasks, with a 1400 ms completion time, and a utilization of 87% for DQL it was 64% utilization, 640 ms completion time. This might be because the superposition encoding in the quantum computer is based on the 'allocation hypotheses' which

maintain more than one hypothesis alive, rather than discarding the sub-optimal greedy solution; this lowers the chance that the model will come up with a sub-optimal solution.[5]

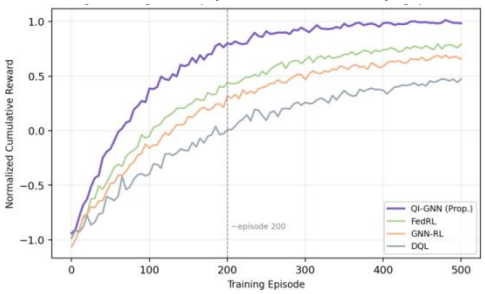


Fig. 3. Cumulative reward over 500 training episodes.

Fig. 3 shows the reward for over 500 training episodes. The reward plateau of QI-GNN is stable and can be achieved after around 200 episodes and retain the advantage over all baseline throughout the training. As for training, the entanglement-regularization term in the reward (Eq. 6) makes it climb more steeply than FedRL or GNN-RL, which is in line with the concept of quantum-enhanced representations having richer gradient signals during training.

F. Efficacy of Quantum-Inspired Encoding

Our hypothesis of encoding is supported by the numbers in Table 4. Representation based on the amplitude (Equations. Each task is represented on the unit circle on the complex space as shown in 4-5, even though the CPU and memory consumption of two tasks is similar, they are represented by different points on the circle when they have different scheduling urgency, which can be aggregated over neighboring tasks by the GNN.

G. Graph Structure Exploitation

The reduction in completion time (25.7%) shows that the scheduler was aware of the node with the highest completion time for each task prior to scheduling it to it, which was made known to the scheduler by the DAG through GraphSAGE. These flat-feature DRL baselines are not able to access that signal and put tasks without knowing what depends upon them downstream. The latency benefits obtained here compare fairly well with the latency results obtained by Rajammal and Chinnadurai [7] for graph based scheduling and by Tung et al. [8] for heterogeneous computing.

H. Energy Efficiency

The model already exhibits packing effect and the current utilization rate is 88%, which means that more machines can be turned off for executing the same tasks and the energy penalty term in the reward (Eq. 6) will be lower and the packing effect will be further boosted. Likewise, Khan et al. [5] report energy savings in the context of quantum-inspired fog scheduling and FedRL [6] is somewhat further removed than energy, since it is not its reward function priority, more towards privacy and communication cost.

VII. CONCLUSION

This paper presents QI-GNN, a resource allocation framework for IoT-cloud networks combining quantum-inspired amplitude encoding, GraphSAGE-based topology learning, and actor-critic reinforcement learning. On the Google Cluster Trace v3 dataset with 1000 concurrent tasks, it reached

88.0% utilization, 275 ms completion time, 4.90 kWh energy consumption, 198 tasks/sec throughput, and a 5.5% SLA violation rate, improvements of 11.4%, 25.7%, 21.0%, 11.2%, and 42.7% over the strongest baseline, FedRL. These gains held across workloads from 200 to 1400 tasks with no sign of dropping off at higher load, which points to an architecture that scales as IoT deployments grow. In practice, this means that if an SLA violation is violated in a real IoT deployment, a failure of the service, and not a delay that can be added as a feature at a later date.

References

[1] Ahanger, Tariq, Fadl Dahan, Usman Tariq, and Imdad Ullah. "Quantum Inspired Task Optimization for IoT Edge Fog Computing Environment." *Mathematics* 11, no. 1 (2022): 156. https://doi.org/10.3390/math11010156.

[2] Chen, Tianrui, Xinruo Zhang, Minglei You, Gan Zheng, and Sangarapillai Lambotharan. "A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks." *IEEE Internet of Things Journal* 9, no. 3 (2022): 1712–24. https://doi.org/10.1109/JIOT.2021.3091551.

[3] Hu, Han, Dingguo Wu, Fuhui Zhou, Xingwu Zhu, Rose Qingyang Hu, and Hongbo Zhu. "Intelligent Resource Allocation for Edge-Cloud Collaborative Networks: A Hybrid DDPG-D3QN Approach." *IEEE Transactions on Vehicular Technology* 72, no. 8 (2023): 10696–709. https://doi.org/10.1109/TVT.2023.3253905.

[4] Ji, Maoxin, Qiong Wu, Pingyi Fan, et al. "Graph Neural Networks and Deep Reinforcement Learning-Based Resource Allocation for V2X Communications." *IEEE Internet of Things Journal* 12, no. 4 (2025): 3613–28. https://doi.org/10.1109/JIOT.2024.3469547.

[5] Khan, Sonia, Naqash Younas, Wahib Jamal Khan, Musaed Alhussein, Khursheed Aurangzeb, and Muhammad Shahid Anwar. "Quantum Inspired Adaptive Resource Management Algorithm for Scalable and Energy Efficient Fog Computing in Internet of Things (IoT)." *Computer Modeling in Engineering & Sciences* 142, no. 3 (2025): 2641–60. https://doi.org/10.32604/cmes.2025.060973.

[6] Mali, Saroj, Feng Zeng, Deepak Adhikari, et al. "Federated Reinforcement Learning-Based Dynamic Resource Allocation and Task Scheduling in Edge for IoT Applications." *Sensors* 25, no. 7 (2025): 2197. https://doi.org/10.3390/s25072197.

[7] Rajammal, K., and M. Chinnadurai. "Dynamic Load Balancing in Cloud Computing Using Predictive Graph Networks and Adaptive Neural Scheduling." *Scientific Reports* 15, no. 1 (2025): 22181. https://doi.org/10.1038/s41598-025-97494-2.

[8] Xuan Tung, Nguyen, Le Tung Giang, Bui Duc Son, et al. "Graph Neural Networks for Next-Generation-IoT: Recent Advances and Open Challenges." *IEEE Communications Surveys & Tutorials* 28 (2026): 2226–62. https://doi.org/10.1109/COMST.2025.3613845.

[9] Walia, Guneet Kaur, Mohit Kumar, and Sukhpal Singh Gill. "AI-Empowered Fog/Edge Resource Management for IoT Applications: A Comprehensive Review, Research Challenges, and Future Perspectives." *IEEE Communications Surveys & Tutorials* 26, no. 1 (2024): 619–69. https://doi.org/10.1109/COMST.2023.3338015.

[10] Zhang, Zhen, Chen Xu, Shaohua Xu, Long Huang, and Jinyu Zhang. “Towards Optimized Scheduling and Allocation of Heterogeneous Resource via Graph-Enhanced EPSO Algorithm.” *Journal of Cloud Computing* 13, no. 1 (2024): 108. https://doi.org/10.1186/s13677-024-00670-4.