International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05257
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

A Review of Graph Neural Networks for In-Network Computing of Real-Time Metaverse Tasks

Likhith Rao K[1], Pradeep Nayak[2], Maithri[3], Manya[4] ,Mithun H[5]

[1] Student, Information Science & Engineering, AIET, Karnataka, India
[2] Assistant Professor, Information Science & Engineering, AIET, Karnataka, India
[3] Student, Information Science & Engineering, AIET, Karnataka, India
[4] Student, Information Science & Engineering, AIET, Karnataka, India
[5] Student, Information Science & Engineering, AIET, Karnataka, India

ABSTRACT

The realization of the "Metaverse"—a persistent, parallel 3D virtual world—depends on the network's ability to deliver immersive experiences without motion sickness. This requires meeting the strict motion-to-photon latency constraint, typically under 20 milliseconds. The reviewed paper by Rashid et al. argues that current network architectures are insufficient for this requirement. Cloud servers incur large propagation delays, while Mobile Edge Com-puting (MEC) servers face heavy queuing delays during large-scale virtual events such as Metaverse concerts.

To address this, the authors propose the use of Computing in the Network (COIN), where programmable switches participate in computation. Rendering tasks are modeled as fork-joint structures and partially processed within the network. The key innovation is replacing slow optimization solvers with a fast Graph Neural Network (GNN) for real-time task placement.

Keyword : - *Metaverse, Graph Neural Network, In-Network Computing, Real Time Task Placement, Mobile Edge Computing*

1. System Model and Mathematical Formulation

The network is modeled as a Software-Defined Network (SDN) consisting of user devices (WDs), in-network computing (INC) nodes, and MEC servers.

1.1 Fork-Joint Task Model

To handle complex rendering workloads, tasks are structured using a "fork-joint" model. A primary rendering task is decomposed (forked) into parallel subtasks which are processed across different network nodes. Once processed, these subtasks are merged (joint) to form the final output. The system performance is critically dependent on the slowest subtask, which defines the overall delay.

1.2 Queuing and Latency Modeling

Network congestion is a primary challenge in the Metaverse. The authors address this by applying an M/M/1 queuing model. The total latency (T_{total}) for any given task is defined as the sum of three components:

$$T_{total} = T_{trans} + T_{exec} + T_{queue}$$

where

- T_{trans} = transmission delay,
- T_{exec} = execution delay,
- T_{queue} = waiting time at the node.

1.3 Optimization Formulation (ILP)

The problem of assigning tasks to specific nodes is formulated as an Integer Linear Programming (ILP) challenge. The objective is to minimize a weighted cost function of system latency and energy consumption. The formulation adheres to several critical constraints:

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 12 | Dec – 2025 DOI: 10.55041/ISJEM05257
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Exclusivity Constraint
A subtask can only be assigned to one node:

$$\sum_{i\in E} X_{1i} \le 1$$

Motion-to-Photon Delay

$$X_1 X_2 \bar{C} \le X_1 X_2 M_{\mathrm{del}}$$

Capacity Constraint

$$\frac{C_i}{\bar{C}} - \sum_{r\in R} \alpha_r > 0$$

where X represents binary decision variables and ¯C is average computing capacity.

2. Proposed Solution: GNN-Based Inference

While ILP provides a mathematically optimal solution, it is computationally expensive and too slow for the millisecond-level decisions required in the Metaverse. To overcome this, the authors introduce a Graph Neural Network (GNN) approach.

Why GNN?
Standard neural networks, such as Multilayer Perceptrons (MLP), often fail to capture the complex topological relationships of a network. GNNs, however, utilize a message-passing mechanism that allows nodes to exchange information with their neighbors. This makes the GNN aware of link congestion and local node properties, offering a distinct advantage in network optimization. The GNN is trained offline using optimal labels generated by the ILP solver, allowing it to make rapid inferences online

Input Features
Each node is represented by a feature vector containing:
- computational capacity,
- current queue length,
- task deadline,
- neighbor link bandwidth.

3. Simulation and Performance Evaluation

A Metaverse concert scenario simulates heavy crowds and mobility.

3.1 Simulation Parameters

Parameter	Value / Description
Network Topology	SDN-based COIN-MEC (12–30 nodes)
User Mobility Model	Adaptive Lévy Walk
Task Model	Fork-Joint Subtask Graph
Latency Bound (M_{del})	50–150 ms
Optimization Solver	Gurobi 11.0
Arrival Distribution	Poisson Process (λ)

3.2 Comparati

Placement accu

Model	Placement Accuracy	Topology Adaptability
Proposed GNN	97.00%	High
Multilayer Perceptron (MLP)	76.25%	Low
Decision Trees (DT)	70.16%	Low

3.3 Key Finding

- Accuracy: The GNN achieves near-optimal ILP-level accuracy (97%).
- Speed: GNN inference is ∼100× faster than ILP, enabling real-time use.
- Scalability: Handles high traffic by offloading tasks to in-network computing nodes

4. Critical Analysis and Conclusion

Strengths
- Strong mathematical modeling using queuing theory.
- GNN naturally fits graph-structured network data.

• Offline ILP training + online GNN inference is practical for real-time constraints.

Weaknesses
• No hardware-level validation (P4 switches, FPGA).
• Energy consumption analysis is limited compared to latency-focused study.

Conclusion
The paper provides a compelling solution to the Metaverse's latency challenge. By integrating in-network computing with GNN-based optimization, the authors demonstrate a scalable and efficient framework for next generation real-time applications in 6G networks.

5. REFERENCES

[1]. Rashid et al., "Graph Neural Network for In-Network Placement of Real-Time Metaverse Tasks in Next-Generation Networks," arXiv, 2024.
[2] I. Aliyu et al., "Towards In-network Computing for Metaverse: Communication, Computing and Cost Modeling," *Proc. 9th Int. Conf. on Advanced Engineering and ICT-Convergence (ICAEIC)*, 2022.

[3] Y. Tsai, Y. Wang, and M. Ouhyoung, "Affordable System for Measuring Motion-to-Photon Latency of Virtual Reality in Mobile Devices," in *ACM SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, 2017.

[4] W. Jiang et al., "Graph Neural Networks for Routing Optimization: Challenges and Opportunities," *Sensors*, vol. 24, no. 1, 2024.

[5] L. U. Khan et al., "Edge-Enabled Metaverse: The Convergence of Metaverse and Mobile Edge Computing," *IEEE Consumer Electronics Magazine*, 2023.