A Cross-Layer Graph Neural Network for Throughput Prediction in 5G Network Digital Twins

Mohamed Ouaddane
Laboratoire LITAN, ESTIN
RN75, Amizour 06300, Bejaia, Algérie
mohamed.ouaddane@univ-lorraine.fr

Abdessamed Seddiki
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
abdessamed.seddiki@univ-lorraine.fr

Lemia Louail
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
lemia.louail@univ-lorraine.fr

Moufida Maimour
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
moufida.maimour@univ-lorraine.fr

Eric Rondeau
Université de Lorraine, CNRS, CRAN
F-54000 Nancy, France
eric.rondeau@univ-lorraine.fr

Meroua Daoudi
Laboratoire LITAN, ESTIN
RN75, Amizour 06300, Bejaia, Algérie
daoudi@estin.dz

***Abstract*—This paper introduces AFQL-GNN, an Attention-based Flow–Queue–Link heterogeneous temporal Graph Neural Network (GNN) for flow-level throughput prediction in 5G Network Digital Twins (NDTs). The model represents each network snapshot as a typed cross-layer graph of application flows, RLC/MAC queues, and wireless links, and combines relation-specific message passing with temporal and within-queue attention to capture how traffic, buffering, scheduling, and radio conditions jointly shape throughput. Evaluated on Simu5G-generated traces spanning heterogeneous traffic regimes, mobility patterns, scheduling policies, queue capacities, transmit powers, and multi-cell configurations, AFQL-GNN achieves an overall Mean Absolute Error (MAE) of 0.79 kbps and consistently outperforms Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and homogeneous GNN baselines across the evaluated scenarios. Results further show low prediction errors across scheduling, transmit-power, and queue-size variations, supporting cross-layer graph learning for throughput estimation in 5G NDTs.**

***Index Terms*—Network Digital Twin, 5G Networks, Graph Neural Networks, QoS, Throughput Prediction.**

I. Introduction

NDTs provide predictive digital representations of communication networks to support rapid what-if analysis and network management [1]. In 5G Radio Access Networks (RANs), flow-level throughput depends on the joint evolution of traffic demand, radio conditions, scheduling, retransmissions, and RLC/MAC queue dynamics. Analytical approaches based on queueing theory [2], [3] rely on simplifying assumptions that may not hold under dynamic 5G conditions, whereas packet-level simulators such as Simu5G and 5G-LENA capture these interactions with high fidelity [4], [5]. However, repeatedly executing such simulations for different traffic profiles, mobility patterns, or network configurations may be computationally expensive, limiting their use for rapid NDT queries.

Data-driven models provide faster alternatives to packet-level simulation. MLP-based predictors learn direct mappings between network measurements and performance indicators [6], while recurrent and attention-based architectures capture temporal dependencies for throughput and Quality of Service (QoS) prediction [7], [15]–[18]. However, these models rely on flat or sequential representations and do not explicitly model interactions among network entities. GNNs address this limitation by exploiting network structure. In wired packet networks, RouteNet and RouteNet-Fermi exploit flow–queue–link interactions for packet-level performance prediction [8], [19]. In wireless networks, GNNs have also been widely investigated for RAN modeling and PHY/MAC tasks such as Key Performance Indicator (KPI) estimation, resource allocation, power control, interference management, and scheduling [10]–[14], [21], [22]. More recently, GLANCE and M3Net use graph-based NDTs to emulate network behavior and predict network KPIs for efficient what-if analysis, with M3Net additionally targeting multiple flow-level metrics such as delay, jitter, and packets dropped [9], [20]. However, these approaches do not explicitly represent application flows, RLC/MAC queues, and wireless links together as interacting entities while simultaneously capturing their temporal evolution. Consequently, how radio conditions, buffering, scheduling, retransmissions, and flow competition jointly propagate across protocol layers toward application-level throughput remains insufficiently modeled.

To address this gap, this paper introduces AFQL-GNN, an Attention-based Flow-Queue-Link (AFQL) heterogeneous temporal GNN for 5G throughput prediction in NDT workflows. Its contributions are : (i) formulating 5G throughput forecasting as a cross-layer graph learning problem that jointly represents application flows, RLC/MAC queue states, and wireless-link conditions ; (ii) proposing a heterogeneous temporal GNN with typed message passing, radio-aware link attention, recurrent state refinement, temporal attention, and within-queue flow attention to capture both cross-layer dependencies and short-term network evolution; (iii) evaluating AFQL-GNN on a Simu5G-generated traces covering heterogeneous traffic regimes, mobility patterns, schedulers, queue sizes, transmit powers, and multi-cell configurations, showing its potential as a surrogate for rapid throughput queries in 5G RAN NDT workflows.

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

The remainder of this paper is organized as follows. Section II presents the proposed AFQL-GNN architecture and its cross-layer flow–queue–link graph representation. Section III describes the experimental methodology and discusses the prediction accuracy, and baseline comparison. Finally, Section IV concludes the paper and outlines future research directions.

II. AFQL-GNN Model Architecture

This section presents the AFQL-GNN architecture, including its dynamic cross-layer graph representation, heterogeneous node embedding, relation-specific message passing, temporal and within-queue attention, and final throughput prediction mechanism.

A. Dynamic Cross-Layer Graph Representation

AFQL-GNN is a learned predictive model for estimating per-flow throughput in 5G RANs. At each time step t, the cross-layer network state is represented as a dynamic heterogeneous graph $\mathcal{G}_t = (\mathcal{V}_t, \{\mathcal{E}_t^r\}_{r\in\mathcal{R}}, \mathbf{X}_t)$, where the node set $\mathcal{V}_t = \mathcal{F}_t \cup \mathcal{Q}_t \cup \mathcal{L}_t$ contains flow nodes $\mathcal{F}_t$, queue nodes $\mathcal{Q}_t$, and wireless-link nodes $\mathcal{L}_t$. Node features at snapshot t are denoted by $\mathbf{X}_t$. The graph connects these entities through the following relation types :

$$\mathcal{R} = \{F{\to}Q,\ L{\to}Q,\ Q{\to}L,\ Q{\to}F,\ L{\leftrightarrow}L\}. \tag{1}$$

For each relation type $r \in \mathcal{R}$, the edge set $\mathcal{E}_t^r$ represents the valid dependencies of type r between nodes in $\mathcal{V}_t$. These relation-specific edge sets evolve with changes in the network topology and state, including flow arrivals and departures, variations in User Equipment (UE)–Next Generation Node B (gNB) associations, and time-varying radio conditions. For T consecutive snapshots, the prediction task is :

$$\hat{y}_{f,t+\Delta} = \Phi_\Theta\left(\mathcal{G}_{t-T+1}, \ldots, \mathcal{G}_t, f\right), \qquad f \in \mathcal{F}_t \tag{2}$$

where Φ_Θ is the learned graph predictor with trainable parameters Θ, and $\hat{y}_{f,t+\Delta}$ is the predicted throughput of flow f at horizon $\Delta \geq 0$, provided that f remains active at $t+\Delta$.

B. Heterogeneous State Embedding

Flow, queue, and link nodes have type-dependent input dimensions. AFQL-GNN therefore uses a separate MLP encoder with Gaussian Error Linear Unit (GELU) activations for each node type to project their raw features into a shared D-dimensional latent space. This produces homogeneous-dimensional latent representations while preserving type-specific feature processing. Figure 1 provides an architectural overview of AFQL-GNN.

C. Sequential Relation-Specific Message Passing

The AFQL-GNN model refines the latent states of flow, queue, and wireless-link nodes through K message-passing iterations. At iteration $k \in \{0, \ldots, K-1\}$, the relation operators are applied sequentially as $F{\to}Q$, $L{\leftrightarrow}L$, $L{\to}Q$, $Q{\to}L$, and $Q{\to}F$, so that each operator uses the latest representations produced by the preceding updates within the same iteration.

For the attention-based $F{\to}Q$ and $L{\to}Q$ relations, the message received by a target node v is computed as

$$\mathbf{m}_{v,t}^{r,k} = \sum_{u\in\mathcal{N}_t^r(v)} \mathrm{softmax}_u\left(\frac{(\mathbf{W}_Q^r \mathbf{h}_{v,t}^k)^\top \mathbf{W}_K^r \mathbf{h}_{u,t}^k}{\sqrt{d_h}}\right) \mathbf{W}_V^r \mathbf{h}_{u,t}^k \tag{3}$$

Here, $\mathbf{m}_{v,t}^{r,k}$ denotes the message received by target node v at snapshot t and message-passing iteration k for relation $r \in \{F{\to}Q, L{\to}Q\}$. $\mathcal{N}_t^r(v)$ is the set of source neighbors of v under relation r, and u denotes a source node in this set. $\mathbf{h}_{v,t}^k$ and $\mathbf{h}_{u,t}^k$ are the latent representations of the target and source nodes, respectively. $\mathbf{W}_Q^r$, $\mathbf{W}_K^r$, and $\mathbf{W}_V^r$ are the trainable query, key, and value projection matrices associated with relation r. The softmax, computed over $u \in \mathcal{N}_t^r(v)$, assigns an attention weight to each source node according to its relevance to v. The symbol $(\cdot)^\top$ denotes transposition, and $d_h = D/H$ is the dimension of each attention head, where D is the shared latent dimension and H is the number of attention heads.

For $F{\to}Q$, each queue attends to its active flows, weighting their contributions according to the queue state. The resulting message is incorporated using a Gated Recurrent Unit (GRU), followed by Layer Normalization (LN) and an Feed-Forward Network (FFN). For $L{\leftrightarrow}L$, Multi-Head Attention (MHA) refines each wireless-link representation using information from other links, capturing shared-resource usage, interference, and correlated radio conditions. A residual connection and LN follow the attention operation. For $L{\to}Q$, the refined wireless-link representations are propagated to their associated queues using the relation-specific attention mechanism in Eq. (3), injecting radio information such as path loss and Signal-to-Interference-plus-Noise Ratio (SINR). The resulting queue state is refined using a GRU, followed by LN and an FFN. For $Q{\to}L$, the updated queue state is propagated back to its link to reflect buffering and service pressure: it is first projected into the link latent space via $\mathbf{W}_L$, then fused with the current link state through a link-specific GRU followed by LN. Finally, $Q{\to}F$ propagates the resulting queue service state to the flows it serves. Rather than broadcasting the same queue representation uniformly, AFQL-GNN uses a flow-dependent gate:

$$\mathbf{m}_{f,t}^{Q\to F,k} = \sigma\left(\mathbf{W}_g\left[\mathbf{h}_{f,t}^k \,\|\, \mathbf{h}_{q(f),t}^k\right]\right) \odot \mathbf{W}_v \mathbf{h}_{q(f),t}^k. \tag{4}$$

Here, $\mathbf{m}_{f,t}^{Q\to F,k}$ is the queue-to-flow message received by flow f, $q(f)$ denotes its serving queue, $\mathbf{h}_{f,t}^k$ and $\mathbf{h}_{q(f),t}^k$ denote the current flow and queue representations, respectively, and $\mathbf{W}_g$ and $\mathbf{W}_v$ are trainable matrices used to compute the gate and transform the queue representation. σ denotes the sigmoid function, $\|$ denotes concatenation, and $\odot$ denotes element-wise multiplication. The gated message is finally incorporated into the flow state using a GRU, followed by LN and an FFN.

D. Temporal and Within-Queue Context

After K iterations, the flow state $\mathbf{h}_{f,t}^K$ encodes the cross-layer context of flow f. To capture short-term temporal dependencies caused by mobility, fast fading, handovers, queue

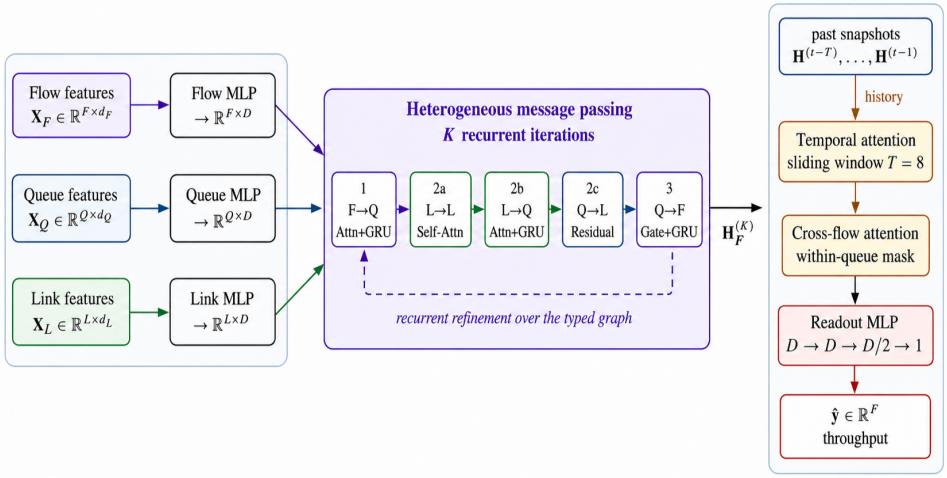


Fig. 1. Overview of the AFQL-GNN architecture.

evolution, and scheduling changes, AFQL-GNN applies MHA over a sliding window of the flow's recent states, with learnable positional embeddings $\mathbf{e}_\tau$:

$$\mathbf{o}_{f,t} = \mathrm{LN}\left(\mathbf{h}_{f,t}^K + \mathrm{MHA}\left(\mathbf{h}_{f,t}^K, \{\mathbf{h}_{f,\tau}^K + \mathbf{e}_\tau\}_\tau\right)\right) \quad (5)$$

The current flow representation acts as the query, while the temporally encoded historical representations provide the keys and values. Flows sharing the same queue compete for a common service process. AFQL-GNN models this via masked MHA, where pairs satisfying $q(g) \neq q(f)$ receive score $-\infty$ before softmax :

$$\mathbf{c}_{f,t} = \mathrm{LN}\left(\mathbf{o}_{f,t} + \mathrm{MMHA}\left(\mathbf{o}_{f,t}, \{\mathbf{o}_{g,t}\}_{g\in\mathcal{F}_t}\right)\right), \quad (6)$$

E. Throughput Prediction and Training Objective

The final representation is mapped by a readout MLP to a standardized throughput prediction $z_{f,t+\Delta} = \rho(\mathbf{c}_{f,t})$. The prediction is then transformed back to the physical throughput domain using statistics computed exclusively from the training partition :

$$\hat{y}_{f,t+\Delta} = z_{f,t+\Delta} \cdot \sigma_Y + \mu_Y, \quad (7)$$

where μ_Y and σ_Y are the mean and standard deviation of training throughput values. Parameters are learned end-to-end by minimizing the empirical prediction error over the training samples and their active flows. Using the Mean Absolute Percentage Error (MAPE) objective, the loss is :

$$\mathcal{L}_{\mathrm{MAPE}}(\Theta) = \frac{1}{|\mathcal{D}|} \sum_{(\mathcal{G}_{t-T+1:t}, y)\in\mathcal{D}} \frac{1}{|\mathcal{F}_t|} \sum_{f\in\mathcal{F}_t} \left| \frac{\hat{y}_{f,t+\Delta} - y_{f,t+\Delta}}{y_{f,t+\Delta} + \epsilon} \right|$$

where $y_{f,t+\Delta}$ is the ground-truth throughput and $\epsilon > 0$ prevents division by zero.

III. Experimental Evaluation

This section presents the experimental workflow for dataset generation, cross-layer graph construction, training, and flow-level throughput evaluation under heterogeneous 5G conditions. It then reports prediction accuracy, baseline comparisons, robustness to cross-layer configuration variations, and runtime efficiency. Figure 2 summarizes the workflow.

TABLE I
Scenario families used to generate the AFQL-GNN dataset.

ID	UEs	gNBs	Regime
SC01	15	1	Small packets
SC02	20	1	Throughput-oriented traffic
SC03	100	1	Mixed service, heterogeneous traffic
SC04	48	4	Mobility stress, fast mobility patterns
SC05	60	4	Cell-edge interference and buffer pressure
SC06	30	2	Controlled protocol/radio variability with heterogeneous traffic.

A. Dataset Generation

We generate the dataset using a packet-level 5G simulation stack built on OMNeT++, INET, and Simu5G, including the Proportional Fair (PF), maximum carrier-to-interference scheduling (MAXCI), Deficit Round Robin (DRR), and MAXCI with multi-band allocation (MAXCI-MB) scheduling disciplines [4]. Simulations combine IP connectivity, 5G radio access, mobility, application traffic, and protocol-layer measurements. Each 120 s run produces snapshots every 0.1 s, from which we extract the cross-layer network state and per-flow throughput labels, yielding over 30 GB of trace data. The radio configuration follows the 3GPP urban macro-cell setting [23], using the `UMa` channel and `P3GPP38901` pathloss model. Across the experiments, the gNB transmit power is varied over $\{0.01, 0.1, 0.5, 2\}$ W and the RLC/MAC queue capacity over $\{50$ KiB, 100 KiB, 2 MiB, 10 MiB$\}$, together with the four scheduling disciplines listed above.

1) Scenario Families: The dataset covers multiple topology scales, mobility patterns, traffic loads, schedulers, and buffer configurations. It includes Ultra-Reliable and Low-Latency Communications (URLLC), enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), mixed-service, and multi-cell scenarios to capture diverse traffic and radio conditions. Table I summarizes the six scenario families.

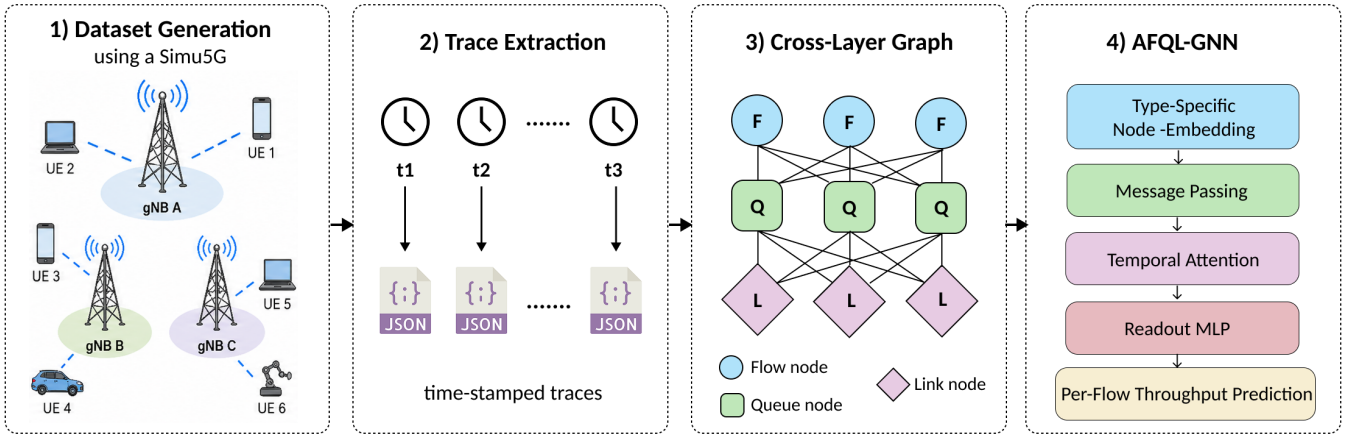


Fig. 2. AFQL-GNN experimental workflow

2) Cross-Layer Snapshot Construction: Each snapshot is converted into the heterogeneous cross-layer graph defined in Section II. Flow nodes encode traffic descriptors, queue nodes capture buffering, scheduling, and service-state information, and wireless-link nodes represent link-quality and mobility-related measurements. Relation-specific edges capture their dependencies, and each active flow is labeled with its measured throughput.

B. Experimental Setup

We use the same AFQL-GNN configuration across all scenarios so that performance differences reflect traffic, mobility, scheduling, queue-size, and radio variations rather than scenario-specific tuning. We use a hidden dimension of $D = 64$, $K = 4$ message-passing rounds, 4 attention heads, a temporal window of $T = 8$ snapshots, temporal attention, and a readout dropout of 0.1. The prediction target is flow-level throughput, with one scalar output per active flow. All experiments use $\Delta = 0$.

For each scenario, simulation runs are split at the run level into 70% training, 15% validation, and 15% test sets, ensuring that snapshots and overlapping temporal windows from the same simulation trajectory cannot appear in different partitions. The validation set is used for model selection, while the test set remains unseen until final evaluation. Prediction accuracy is evaluated using MAE, reported in kbps and defined as $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$, where N is the number of evaluated flow samples, and $\hat{y}_i$ and y_i denote the predicted and ground-truth throughput of sample i, respectively.

C. Results and Discussion

1) Overall Prediction Accuracy: Table II reports the MAE across all scenarios. Overall, AFQL-GNN achieves an average MAE of 0.79 kbps, indicating accurate flow-level throughput estimation across heterogeneous 5G conditions. The error remains low across all scenario families, ranging from 0.01 kbps in SC06 to 2.16 kbps in SC01. In particular, AFQL-GNN maintains low prediction errors in SC03–SC05 despite heterogeneous traffic, mobility, multi-cell operation, interference, and buffer variations.

TABLE II
AVERAGE THROUGHPUT PREDICTION MAE PER SCENARIO.

Scenario	SC01	SC02	SC03	SC04	SC05	SC06	Avg.
MAE (kbps)	2.16	0.13	0.40	0.81	1.22	0.01	**0.79**

TABLE III
THROUGHPUT PREDICTION MAE (KBPS) ACROSS THE SIX SCENARIOS. BOLD: BEST; BLUE: SECOND-BEST.

Model	SC01	SC02	SC03	SC04	SC05	SC06
AFQL-GNN	**2.16**	**0.13**	**0.40**	**0.81**	**1.22**	**0.01**
Homogeneous GNN	7.00	0.40	2.44	6.74	10.80	**0.01**
MLP	12.44	2.41	6.23	38.42	70.16	12.38
LSTM	12.02	3.83	5.81	20.21	19.37	3.70

2) Comparison with Baseline Predictors: We compare AFQL-GNN with three baselines: an MLP using a flat snapshot representation, an LSTM modeling temporal dependencies without the flow–queue–link topology, and a homogeneous GNN treating all node and relation types uniformly. All models use the same data partitions and targets. AFQL-GNN has 270,721 trainable parameters, compared with 36,353 for the MLP, 75,905 for the LSTM, and 71,873 for the homogeneous GNN. Table III reports their MAE across the six scenarios.

AFQL-GNN achieves the lowest MAE in SC01–SC05 and ties with the homogeneous GNN in SC06. The gains are particularly marked in SC04 and SC05, where the MAE drops from 38.42 and 70.16 kbps for the MLP to 0.81 and 1.22 kbps with AFQL-GNN. The homogeneous GNN is the strongest baseline, which confirms the benefit of exploiting graph connectivity. In SC03, AFQL-GNN reduces the MAE from 2.44 to 0.40 kbps, supporting heterogeneous relation-specific message passing.

3) Robustness to Cross-Layer Configuration Variations: We assess AFQL-GNN robustness to scheduling policy, trans-

TABLE IV
AFQL-GNN THROUGHPUT PREDICTION MAE (KBPS) ACROSS SCHEDULING POLICIES, TRANSMIT POWERS, AND QUEUE CAPACITIES.

Variation	Setting	SC01	SC02	SC03	SC04	SC05	SC06
Scheduler	PF	1.84	0.13	0.26	0.74	1.19	0.01
	MAXCI	2.03	0.17	0.63	0.77	1.25	0.01
	DRR	0.40	0.12	0.26	0.76	1.22	0.01
	MAXCI-MB	4.14	0.12	0.48	0.80	1.13	0.01
gNB Transmit power	0.01 W	1.75	0.15	0.53	0.80	1.17	0.01
	0.1 W	1.71	0.13	0.39	0.75	1.21	0.01
	0.5 W	1.73	0.13	0.40	0.76	1.22	0.01
	2 W	1.72	0.13	0.39	0.76	1.19	0.01
Queue size	50 KiB	1.80	0.14	0.59	0.74	1.20	0.01
	100 KiB	2.09	0.13	0.56	0.79	1.23	0.01
	2 MiB	1.44	0.13	0.24	0.77	1.19	0.01
	10 MiB	1.59	0.13	0.22	0.77	1.16	0.01

mit power, and queue size. Table IV summarizes the MAE across all configurations and scenarios. Across scheduling policies, prediction errors remain low overall, with the largest variations observed in SC01 and SC03. Performance is also stable across transmit powers from 0.01 to 2 W, although SC03 exhibits higher error at 0.01 W. Similarly, AFQL-GNN remains accurate across queue capacities, with SC03 showing higher errors for 50 and 100 KiB than for 2 and 10 MiB queues. Overall, the consistently low MAE across the evaluated configurations indicates that AFQL-GNN is robust to scheduling, transmit-power, and queue-size variations.

IV. CONCLUSION

This paper introduced AFQL-GNN, an attention-based heterogeneous temporal GNN for flow-level throughput estimation in 5G NDTs. By representing application flows, RLC/MAC queues, and wireless links as distinct but interacting graph entities, the model captures cross-layer dependencies through relation-specific message passing, temporal attention, and within-queue flow attention.

Evaluated on Simu5G traces covering heterogeneous traffic regimes, mobility patterns, scheduling policies, transmit powers, queue capacities, and multi-cell configurations, AFQL-GNN achieves an overall MAE of 0.79 kbps and consistently outperforms MLP, LSTM, and homogeneous GNN baselines. The gains are particularly pronounced in scenarios involving heterogeneous traffic, mobility, interference, and queue interactions. Moreover, AFQL-GNN maintains low prediction errors across scheduling policies, transmit-power levels, and queue-capacity settings, demonstrating its robustness to cross-layer configuration variations.

Future work will evaluate generalization to unseen topology scales and real-network measurements, investigate future prediction horizons ($\Delta > 0$), conduct component-level ablation studies, extend the framework to additional QoS indicators such as delay, jitter, packet loss, and reliability, and integrate AFQL-GNN into an operational NDT control loop for online what-if analysis and network reconfiguration.

REFERENCES

[1] M. Kherbache et al., "When digital twin meets network softwarization in the industrial IoT: Real-time requirements case study," *Sensors*, 2021.

[2] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. Wiley, 1975.

[3] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, 2001.

[4] G. Nardini et al., "Simu5G: An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," *IEEE Access*, 2020.

[5] N. Patriciello, et al., "An E2E Simulator for 5G NR Networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019.

[6] M. K. Banjanin et al., "Adaptive Modeling of Prediction of Telecommunications Network Throughput Performances in the Domain of Motorway Coverage," *Applied Sciences*, vol. 11, no. 8, p. 3559, 2021.

[7] L. Li et al., "Research on Throughput Prediction of 5G Network Based on LSTM," *Intelligent and Converged Networks*, 2022.

[8] M. Ferriol-Galmés et al., "RouteNet-Fermi: Network modeling with graph neural networks," *IEEE/ACM transactions on networking*, 2023.

[9] B. Li et al., "GLANCE: Graph-Based Learnable Digital Twin for Communication Networks," *arXiv preprint arXiv:2408.09040*, 2024.

[10] Y. Shen et al., "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. on Wireless Com.*, 2022.

[11] T. Chen et al., "A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks," *IEEE IoT Jal*, 2022.

[12] A. Chowdhury et al., "Unfolding WMMSE using graph neural networks for efficient power allocation," *IEEE Trans. on Wireless Com.*, 2021.

[13] Z. Zhao et al., "Link scheduling using graph neural networks," *IEEE Trans. on Wireless Com.*, 2022.

[14] J. Perdomo et al., "WirelessNet: An Efficient Radio Access Network Model Based on Heterogeneous Graph Neural Networks," *IEEE Access*, 2025.

[15] L. Mei et al., "Realtime Mobile Bandwidth and Handoff Predictions in 4G/5G Networks," *Computer Networks*, vol. 204, p. 108736, 2022.

[16] H. Na et al., "LSTM-Based Throughput Prediction for LTE Networks," *ICT Express*, vol. 9, no. 2, pp. 247–252, 2023.

[17] S. Barmpounakis et al., "LSTM-Based QoS Prediction for 5G-Enabled Connected and Automated Mobility Applications," in *Proc. of IEEE 5GWF*, 2021.

[18] S. Mostafavi et al., "Probabilistic Delay Forecasting in 5G Using Recurrent and Attention-Based Architectures," *arXiv preprint arXiv:2503.15297*, 2025.

[19] K. Rusek et al., and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE Journal on Selected Areas in Communications*, 2020.

[20] B. Guda et al., "M3Net: A Multi-Metric Mixture of Experts Network Digital Twin with Graph Neural Networks," *arXiv preprint arXiv:2512.09797*, 2025.

[21] Y. Lu et al., "Graph neural networks for wireless networks: Graph representation, architecture and evaluation," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 150–156, 2024.

[22] Z. Zhao et al., "Delay-Oriented Distributed Scheduling Using Graph Neural Networks," in *Proc. IEEE ICASSP*, 2022.

[23] 3GPP, "Study on Channel Model for Frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project, Technical Report TR 38.901, 2025.