

Neighbor Auto-Grouping Graph Neural Networks for Handover Parameter Configuration in Cellular Network

Mehrtash Mehrabi^{1,2}, Walid Masoudimansour¹, Yingxue Zhang¹, Jie Chuai¹, Zhitang Chen¹, Mark Coates³, Jianye Hao^{1,4}, Yanhui Geng¹

¹ Huawei Noah's Ark Lab, Montreal, Canada

² University of Alberta, Edmonton, Canada

³ McGill University, Montreal, Canada

⁴ Tianjin University, Tianjin, China

{mehrtash.mehrabi, walid.masoudimansour, yingxue.zhang, chuajie, chenzhitang2, haojianye, geng.yanhui}@huawei.com, mark.coates@mcgill.ca

Abstract

The mobile communication enabled by cellular networks is the one of the main foundations of our modern society. The configuration of network parameters is crucial for optimizing the performance of cellular networks and providing better user experience, but it's becoming complicated with the increasing size and complexity of the networks. The current practice relies on experts' prior knowledge, which is not adequate and requires a lot of maintenance costs. We propose a learning-based framework for handover parameter configuration, which addresses the complicated dependencies between neighboring cells and optimizes the whole network. We introduce a novel approach called auto-grouping graph convolutional network (AG-GCN), which imitates how the network responds to different network states and parameter values. Our framework uses a local multi-objective optimization strategy to balance the performance of each cell and its neighbors during the parameter configuration stage. It achieves better average network throughput compared to experts' recommendations and alternative baselines and has the potential to reduce costs arising from human expert intervention and maintenance.

Introduction

The rapid growth in the number of devices that need real time, high quality connection to the internet (e.g., internet of things (IoT) devices, health monitoring equipment, devices used for online education and remote working, autonomous vehicles, etc.) makes it essential to improve cellular network performance. Unsatisfactory user experience and network interruption have negative impacts in our modern society. Thus, improving the cellular network has both economic and social impact towards achieving United Nations Sustainable Development Goals (UNSDGs) (United Nations 2015; World Economic Forum 2020). Moreover, it can highly contribute to enhancing infrastructure, promoting sustainable industrialization, fostering innovation, responsible consumption, enabling sustainable cities and communities, and promoting decent work and economic growth (Gohar and Nencioni 2021; Rao and Prasad 2018; Siriwardhana et al. 2021).

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The performance of a cellular network relies heavily on its parameter configurations and it is becoming more crucial, as the number of mobile users continues to grow rapidly (statista 2022). These parameters govern access control, handover, and resource management (Dahlman, Parkvall, and Skold 2013; Bhat et al. 2012). One of the parameters that has a significant impact on the quality of service (QoS) in such networks is the handover (Tekinay and Jabbari 1991).

Optimizing handover parameters is one of the most common approaches to guarantee minimum service delay or interruption and improve coverage and throughput (Mu, Barco, and Fortes 2014). However, with the massive increase in both the size and complexity of cellular networks, parameter configuration is becoming complicated. The current practice, which relies largely on experts' prior knowledge, is inadequate, requiring many domain experts and leading to high maintenance costs.

One of the key challenges in the network parameter optimization problem is the complex spatial and temporal dependencies in the cellular network. Any employed algorithm should be capable of tracking the non-stationary changes in the environment, i.e., the fluctuations of user number, network load, etc. (Agiwal, Roy, and Saxena 2016). Also, due to the diverse characteristics of cells across the network, the best parameter configuration for one cell may not be optimal for another and parameter configuration of one cell not only affects its own performance, but also affects its neighbors' (Dahlman, Parkvall, and Skold 2013). Therefore, there are strong interactions between neighboring cells which become extremely complicated in heterogeneous network. Consequently, developing an algorithm that can adapt to the temporal dynamics and cell diversity in real networks is essential for parameter configuration (Jiang et al. 2016).

The current cellular network deployments rely heavily on human-designed rules and analytical models based on limited network states and parameters, which cannot fully capture the complex relationships and non-stationary changes in network dynamics. These models may also be too simplified, leading to degraded performance, and unable to deal with the diverse cells in the network, making them sub-optimal (Imran, Zoha, and Abu-Dayya 2014).

Recently, data-driven approaches based on machine learn-

ing (ML) have been extensively used for parameter configuration and network management in cellular networks (Yu et al. 2017; Tabrizi, Farhadi, and Cioffi 2012; Riihijarvi and Mahonen 2018; Chuai et al. 2019; Ye et al. 2013). The multi-layer perceptron (MLP) has been identified as a universal function approximator (Goodfellow, Bengio, and Courville 2016), making it suitable for environments like cellular networks that lack an accurate analytical model and are highly dynamic. ML models are capable of utilizing high-dimensional information and approximating complex functions, allowing for a more accurate prediction of network performance metrics and parameter configurations, which cannot be achieved by human experience.

In order to address the above challenges, we investigate two important questions: 1) **Modeling**: how to model the spatial and temporal dependencies of the cellular network? 2) **Decision-making**: how to choose the parameter values to jointly optimize the overall performance of interconnected and interacting cells? We first, propose a ML-based model to precisely imitate the cellular network environment and then, use it to configure the parameters.

We demonstrate that the proposed handover parameter configuration framework can improve the average network throughput compared to existing methods while reducing costs from human expert intervention and maintenance. This has the potential to provide better internet access to underserved areas and enable important applications like online education and health monitoring (Attaran 2021). Our main contributions are summarized as follows.

- We propose a novel method to model the impact from the neighbors of each cell in a distinguishable way to capture the complex spatial dependencies of the network.
- We consider the changing dynamics of the network in our reward model to better reflect the temporal dependencies.
- We introduce a multi-objective optimization strategy based on the model to consider several performance metrics and improve the overall network throughput, which has the potential for high social impact applications.

Background and Related Work

The adjustment of handover parameters can significantly affect network throughput and traffic load balance. During the handover process, the user equipment (UE) monitors the received signal power (RSRP) of the serving cell and reports measurements to prepare for handover when the RSRP drops below a pre-defined threshold (A2-threshold) (3GPP TS36.331 2016). Increasing the A2-threshold decreases the number of UEs triggering handover in the serving cell, spreading load to its neighbors and impacting throughput. Decreasing the A2-threshold may cause poor experience for edge UEs and repeated connection loss. Techniques such as fuzzy systems, deep reinforcement learning (DRL), and contextual bandit have been used to optimize wireless network parameters.

The use of graph convolutional networks (GCNs) (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2017; Fan et al. 2019) has also yielded significantly well-designed

models to predict the network traffic and optimize the corresponding parameters. For example, in (Zhang et al. 2020), the authors introduce a novel handover strategy based on GCNs. The handover process is modeled as a directed graph by which the user tries to predict its future signal strength. Other works such as (Zhao et al. 2020) introduce novel methods of network traffic prediction combined with a greedy search or action configuration method to optimize handover parameters. However, these works fail to consider the heterogeneous aspect of the cellular networks.

Despite being effective, none of the above-mentioned methods uses the capacity of the neighbors' information to fully tailor the model to adapt to the spatial characteristics of a cellular network, where the interaction is complex and the network is heterogeneous. Also, despite the fact that these techniques consider some important measures of optimization, none of them approaches the problem at hand by considering two of the most important measures simultaneously (especially from the users' perspective): load balancing and throughput. In this article, we propose an effective and efficient framework that models the network as a heterogeneous graph where we learn an implicit interaction type for each neighboring cell. Then, it incorporates the impact of neighboring cells from each interaction group in a unique way. Moreover, in contrast to the available methods in the literature, we exploit two important measures in the network simultaneously, to configure the parameters effectively: throughput and load balancing, which are directly related to the user experience in the network.

Problem Formulation

Let us consider a network with N cells, and form N clusters each composed of one of the network cells as its center cell along with its neighboring cells. As an example, we choose the optimization of the A2-threshold to investigate the performance of our algorithm. According to the 3GPP standard (3GPP TS36.331 2016), an A2 event is triggered when the received power at user u from cell n , $P^{u,n}$, satisfies

$$P^{u,n} + H_{ys} < Thresh, \quad (1)$$

where H_{ys} is the hysteresis parameter to avoid frequent handovers and $Thresh$ is the A2-threshold we are optimizing.

We consider an online optimization process. In real practice, network operators are often conservative and only allow a limited number of experiments. During the optimization period of L days, and the A2-threshold can be adjusted once for each cell at the beginning of each day. For day t , let D_t be the total bits transmitted by all the cells, and T_t be the total transmission time. We would like to maximize the accumulated network throughput of the optimization period, i.e., $\max \sum_{t=1}^L \frac{D_t}{T_t}$. Maximizing the overall network throughput by jointly optimizing the A2-threshold of all cells is difficult. The problem becomes even more complicated as the network size increases, which makes a centralized solution not scalable. The adjustment of the A2-threshold of one cell only affects its local neighborhood and thus, we convert the centralized problem into a local decision problem. That is, each cell only examines its local performance metrics and chooses its own parameter configuration value.

The adjustment of the A2-threshold affects the network throughput via two means: better resource utilization by load balancing, and improved cell throughput with less connection loss and measurement reporting. Consequently, in order to configure it, these two metrics must be considered in the local decision problem. The throughput of cell i on day t is highly dependent on its A2-threshold, formulated as action a_t^i , denoted as $\alpha_t^i(a_t^i)$. The load balancing factor in the i -th cluster with center cell i on day t with a_t^i is defined as the ratio of the center cell throughput to the average throughput of its neighboring cells, denoted by $\beta_t^i(a_t^i)$ and formulated as $\beta_t^i(a_t^i) = \alpha_t^i(a_t^i) / \bar{\alpha}_t^i$, where $\bar{\alpha}_t^i$ is the average throughput of the neighbors of cell i with action a_t^i and, denoting by $\mathcal{N}_t(i)$ the set of all neighbors of cell i on day t , it can be formulated as $\bar{\alpha}_t^i = \frac{1}{|\mathcal{N}_t(i)|} \sum_{j \in \mathcal{N}_t(i)} \alpha_t^j(a_t^j)$. The throughput ratio (rather than traffic/user ratio) is used since different cells have different capacities. This value approaches 1 when loads of different cells match their capacities.

Our goal is to maximize the overall network throughput on each day by optimizing the two important network performance metrics, namely, throughput ratio $\beta_t^i(a_t^i)$ and cell throughput $\alpha_t^i(a_t^i)$ for each cell $i \in [1, N_t]$, where N_t is the total number of cells on day t , at the same time. Therefore, we propose the following optimization problem for tuning the A2-threshold for cell i :

$$\arg \max_{a_t^i \in \mathcal{A}} \left(-\sqrt{|1 - \beta_t^i(a_t^i)|}, \alpha_t^i(a_t^i) \right), \quad (2)$$

where \mathcal{A} is the set of all possible values for the A2-threshold in the cellular network.

The challenge of solving the above problem lies in several folds. *First*, since the network performance function is complex, dynamic and unknown, obtaining accurate $\beta_t^i(a_t^i)$ and $\alpha_t^i(a_t^i)$ is difficult. Instead, in this work, we adopt a data-driven approach to learn reward models and estimate the performance metrics. *Second*, in real-world cases, only a limited experimental budget is allowed by network operators leading to insufficient diverse historical data (state, action pairs) to train a data-driven learning model. In our design, we use a data augmentation technique in the form of neighbor cell augmentation to enrich the features from each cell. *Third*, the handover parameter configuration is affected by adjacent cells. Thus, it is essential to model the information coming from the adjacent cells to achieve accurate reward modeling. *Lastly*, optimizing one performance metric greedily might hinder another, thus, how to jointly optimize different performance metrics needs careful consideration.

Temporal Auto-Grouping GCN for Reward Modeling

In order to better capture the dependency between each cell and its neighboring cells, we *first* introduce our novel method for neighboring cell feature aggregation. *Second*, we propose a temporal feature aggregation step with recurrent neural networks (RNN) to model the temporal correlation from the historical sequence of the network states. *Third*, we elaborate the overall training process, considering the im-

pact from the neighboring cells, the temporal correlation in the network and the action we aim to optimize.

Spatial Feature Modeling

The handover parameters are important for the learning problem in both the center and neighboring cells in cellular networks. We propose to capture the neighboring cell information by using message-passing neural networks (MPNNs), specifically graph neural networks (GNNs). GNNs are effective in modeling real-world applications with structural information by incorporating each node's features and those of its neighboring nodes in each layer (Hamilton, Ying, and Leskovec 2017; Ying et al. 2018; Wang et al. 2019). We believe that the GNN framework is well-suited for handling the dependencies between the center cell and its neighboring cells in cellular networks.

Graph-Based Cellular Network Modeling We construct a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$ for day t , where each node $v \in \mathcal{V}_t$ represents one cell and is associated with a feature vector $\mathbf{x}_t^v \in \mathbb{R}^d$ (v -th column of $\mathbf{X}_t \in \mathbb{R}^{d \times |\mathcal{V}_t|}$), including the statistical properties of node v measured on day t . The statistical properties could include several features such as the antenna transmission power, physical resource block (PRB) usage ratio, the amount of data traffic, and the transmission bandwidth. These features serve as the node attributes. The edge set \mathcal{E}_t encodes the interactions between cells based on the handover events between pairs of cells. Based on historical data, if any pair of cells has an average number of handover events above a threshold τ , we assume an edge between those two cells. The neighboring set for node v is denoted as $\mathcal{N}_t^g(v) = \{u | u \in \mathcal{V}_t, (u, v) \in \mathcal{E}_t\}$.

Due to the heterogeneous nature of the cellular network, the relationships between the neighboring cells can be complex. Concretely, there might be an implicit M latent relationship types $\mathcal{R} = \{r_1, r_2, \dots, r_M\}$ that can be learned to better handle the complex interactions in the cellular networks. The proposed auto-grouping GCN (AG-GCN) approach aims to capture the interactions between neighboring cells in a distinguishable way. It does so by dividing neighboring cells into different groups, where each group provides shared information. AG-GCN is inspired by a recent work (Pei et al. 2020) and aims to handle the unique properties of cellular networks. The approach is described in detail in the following sections

Neighborhood Augmentation The limited experiment budget in cellular network modeling leads to a lack of diverse historical data. In addition, constructing the graph based on handover events may result in cells with a limited number of neighboring cells. To address these issues, we use a data augmentation technique to enrich the features of each cell based on the similarity between cells in a latent space.

We define a feature transformation function $f(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^l$ which maps the input node feature $\mathbf{x}_t^v \in \mathbb{R}^d$ to a latent space $\mathbf{y}_t^v = f(\mathbf{x}_t^v) \in \mathbb{R}^l$. In order to capture the long-range dependencies and similarity in the cellular network, we design an additional neighborhood in the latent representation space based on Euclidean distance. For each node $v \in \mathcal{V}_t$, we form the augmented neighborhood $\mathcal{N}_t^l(v) =$

$\mathcal{N}_t^g(v) \cup \mathcal{N}_t^s(v)$, where $\mathcal{N}_t^g(v)$ and $\mathcal{N}_t^s(v)$ are the neighbors of node v in the original graph and in the latent space, respectively. The neighbors in the latent space are selected based on their Euclidean distance to the center cell. The n nearest nodes in the latent space are selected to create $\mathcal{N}_t^s(v)$ for cell v , where the number of nodes we select based on the feature similarity is equal to the neighborhood size in the original graph $|\mathcal{N}_t^g(v)|=|\mathcal{N}_t^s(v)|=n$. The neighbor augmentation module in Fig. 1 illustrates this process.

Neighborhood Auto-Grouping Once we have obtained the augmented neighborhood set, the neighbors in the augmented neighborhood $\mathcal{N}_t(v)$ are divided into different groups by a geometric operator γ . Consider node v and its neighbor node $u \in \mathcal{N}_t(v)$. The relation between them on day t is denoted as $\gamma(\mathbf{y}_t^v, \mathbf{y}_t^u) : (\mathbb{R}^l, \mathbb{R}^l) \rightarrow \mathcal{R} = \{r_1, r_2, \dots, r_M\}$. This grouping aims at combining neighbors' information in groups with similar inter-group features. For each group $r_i \in \mathcal{R}$, the neighborhood feature set on day t is defined as $\mathcal{N}_t^{r_i}(v) = \{u | u \in \mathcal{N}_t(v), \gamma(\mathbf{y}_t^v, \mathbf{y}_t^u) = r_i\}$. The auto-grouping module in Fig. 1 demonstrates this process. Note that yellow neighbors (marked with *) are the projected counterparts of the neighbors in the graph space, while the green neighbors (marked with +) correspond to the augmented neighbors from the latent space.

Conditional Message Passing Since the order within each neighbor group should not impact the output of the representation, we apply a permutation invariant function $\pi(\cdot)$ on the neighbors within each group (mean pooling across each feature dimension) and aggregate them separately. Fig. 1 shows an example of the AG-GCN, where $l = 2$ and $|\mathcal{R}| = 4$ and the representation after the permutation invariant function $\pi(\cdot)$ is shown by black dashed arrows ended to nodes 1, 2, 3, and 4. Then for each group $r_i \in \mathcal{R}$, a non-linear transform is further applied as:

$$\mathbf{z}_t^{v,r_i} = \sigma\left(\mathbf{W}_t^{v,r_i} \cdot \pi(\{\mathbf{x}_t^u | u \in \mathcal{N}_t^{r_i}(v)\})\right), \quad (3)$$

where \mathbf{W}_t^{v,r_i} is a learnable weight matrix for the neighbors in group r_i of node v on day t , and $\sigma(\cdot)$ is a non-linear function, e.g., tanh. Then for each node we aim to aggregate the transformed neighborhood features from their different groups of neighbors in a distinguishable way. The vectors \mathbf{z}_t^{v,r_i} for $r_i \in \mathcal{R}$ are further aggregated as $\mathbf{h}_t^v = [\mathbf{z}_t^{v,r_1}; \dots; \mathbf{z}_t^{v,r_M}]$, where $[\ ;]$ represents concatenation.

Temporal Feature Modeling

We propose to use additional temporal features for each center cell to extract the changing dynamic pattern of its states within each day to further improve the reward model performance. We assume the samples of the center cell v on day t can be divided into K groups by their temporal order. For all the samples in each group k , we take the average network state for each group of samples and denote it as $\mathbf{x}_{t,k}^v$. We use an RNN layer to capture this temporal dependency of the features from different groups by feeding all the network states as an input sequence $\mathbf{P}_t^v = [\mathbf{x}_{t,1}^v; \mathbf{x}_{t,2}^v; \dots; \mathbf{x}_{t,K}^v]^T \in \mathbb{R}^{K \times d}$, to obtain $\mathbf{c}_t^v =$

$\text{RNN}(\mathbf{P}_t^v, \delta) \in \mathbb{R}^{d'}$, where δ and d' are the set of trainable parameters and the output dimension of the RNN layer, respectively.

Overall Training Pipeline

The model aims to predict the throughput ratio and throughput of a cellular network's center cell for the next day based on current network states and actions taken. The performance metrics are influenced by the actions taken and previous performance metrics. Hence, we consider the current throughput ratio, i.e., β_t^v , in the prediction process.

To make the final prediction, the learned representation of the neighborhood by the AG-GCN aggregation, the temporal features of the center cell, and the throughput ratio of the current day, i.e. β_t^v , are concatenated to form the state vector of cell v as $\mathbf{s}_t^v = \Psi(\mathbf{W}_t^v \cdot [\beta_t^v; \mathbf{c}_t^v; \mathbf{h}_t^v])$, where \mathbf{W}_t^v is a learnable weight matrix for node v on day t , and $\Psi(\cdot)$ is a non-linear function, e.g., tanh. Since the final representation should be sensitive to the chosen input action (of which the decision making process will be elaborated later), the throughput ratio and throughput of the next day for cell v are formulated as the output of a non-linear transformation $\Lambda(\cdot)$ function of state and action:

$$\hat{\beta}_{t+1}^v = \Lambda(\mathbf{W}_\beta^v \cdot ([\mathbf{s}_t^v; a_t^v])), \quad (4)$$

$$\hat{\alpha}_{t+1}^v = \Lambda(\mathbf{W}_\alpha^v \cdot ([\mathbf{s}_t^v; a_t^v])), \quad (5)$$

where \mathbf{W}_β^v and \mathbf{W}_α^v are trainable matrices of node v for throughput ratio and throughput models, respectively. The overall flow of data from the graph structure to the final prediction is represented in Fig. 1. Note that we train two separate models for predicting the throughput and the throughput ratio simultaneously.

To properly use the A2-threshold for the prediction, we use the change in this parameter compared to the previous day as the action $a_t^v = A2_{t+1}^v - A2_t^v$, where $A2_{t+1}^v$ and $A2_t^v$ are the A2-thresholds for cell v on day $t+1$ and t , respectively. The reason for this design choice has twofold. First, it reduces the range of the action space and makes it easier for the model to learn. Besides, the delta action directly reflects the change in the cell coverage/loads, so they are more sensitive to the performance metrics. To form the training objective, we consider data of $T+1$ consecutive days and form the pairs $(t, t+1)$, $t \in \{1, 2, \dots, T\}$, to predict the throughput ratio and throughput of the center cell in day $T+1$, by minimizing the following loss functions respectively:

$$\frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{v=1}^{N_t} (\hat{\beta}_{t+1}^v - \beta_{t+1}^v)^2 + \lambda_1 \|\Theta_1\|^2, \quad (6)$$

$$\frac{1}{T} \sum_{t=1}^T \frac{1}{N_t} \sum_{v=1}^{N_t} (\hat{\alpha}_{t+1}^v - \alpha_{t+1}^v)^2 + \lambda_2 \|\Theta_2\|^2, \quad (7)$$

where λ_1 and λ_2 are the hyperparameters chosen for regularization. Θ_1 and Θ_2 represent all the trainable parameters in the models. The trained reward model is now able to mimic the real network and predict both throughput ratio and throughput of each center cell for the coming day and can be used to check the impact of actions towards the performance metrics we are considering.

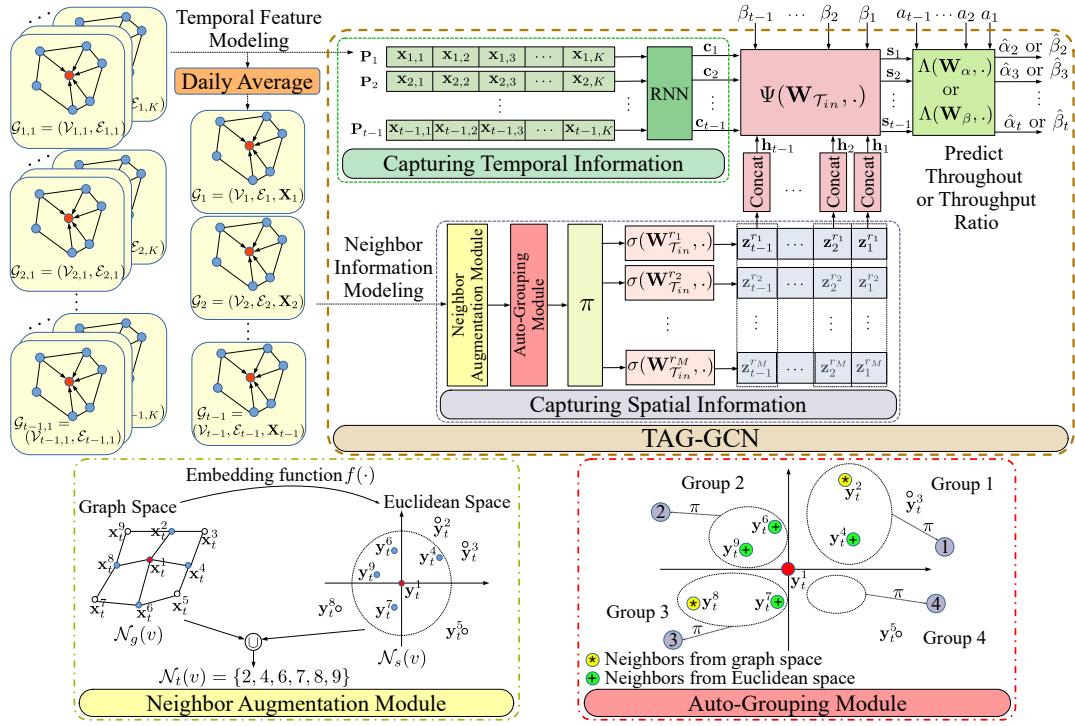


Figure 1: The flow of information from graph structure to final prediction, used to form the training pipeline of two models for predicting the throughput \hat{a} and the throughput ratio $\hat{\beta}$ for $T_{in} = \{1, 2, \dots, t-1\}$. In this demo example, the auto-grouping module constructs $M = 4$ groups of neighbors, where $l = 2$. Empty groups are filled with the average of other groups.

Action Configuration

As discussed in the earlier sections the main objectives to consider in the action configuration process are load balancing, identified by the throughput ratio, and the cell throughput. Hence, the best action for cell v on day t , i.e. $a_t^v \in \mathcal{A}$, is the one that optimizes the problem in (2). In general, when dealing with a multi-objective problem, different objectives are often conflicting, and we may not be able to optimize them simultaneously. One common way to tackle this problem is to give different objectives weights and optimize the weighted objective value. However, in our scenario, it is difficult to determine the weights and different clusters may require cluster-specific weights. Here we break the problem in (2) into two sub-problems, and solve them sequentially. We first optimize the action with respect to the predicted throughput ratio, i.e., $\hat{\beta}_{t+1}^v(a_t^v)$ for cell v on day t , where $a_t^v \in \mathcal{A}$, and then optimize the throughput $\hat{a}_{t+1}^v(a_t^v)$. Specifically, the throughput ratio is optimized and we find the set of best c values for a_t^v , denoted \mathcal{A}_c^v , such that

$$\min_{a_t^v \in \mathcal{A}_c^v} -\sqrt{|1 - \hat{\beta}_{t+1}^v(a_t^v)|} \geq \max_{a_t^v \in \mathcal{A} - \mathcal{A}_c^v} -\sqrt{|1 - \hat{\beta}_{t+1}^v(a_t^v)|}. \quad (8)$$

Then, our goal is to achieve the maximum possible throughput for cell v on day t and this is through

$$\hat{a}_t^v = \arg \max_{a_t^v \in \mathcal{A}_c^v} \hat{a}_{t+1}^v(a_t^v). \quad (9)$$

\hat{a}_t^v is then the final recommended action for cell v on day t . This procedure for all the N_t cells of the network on day t is

presented in Algorithm 1.

Experimental Results

The experiments are conducted on a large-scale cellular network simulator constructed from real-world data. Principal Component Analysis (PCA) is used as the mapping function $f(\cdot)$ to obtain a 2-dimensional latent representation in the AG-GCN step. This transformation enables neighborhood augmentation and neighbor group assignment. Then, the relationship operator γ assigns a group to each subset of points in each quadrant of the 2-dimensional space, as presented in Table 1. The permutation-invariant function π applied to each group of neighbors is average in the experiments.

Datasets

To perform our experiments and evaluate the proposed model, two datasets are used in this study:

Dataset-A: A real metropolitan cellular network containing around 1500 cells sampled hourly and collected from Oct. 17 to Oct. 31, 2019. Each data sample contains information such as the cell ID, sample time, configuration of cell parameters, and measurements of the cell states.

Dataset-B: Also a real metropolitan cellular network. The network contains 1459 cells, and the data is collected from Sep. 1 to Sep. 29, 2021. Each data sample contains similar information as above.

Algorithm 1: TAG-GCN for Action Configuration

Input: \mathbf{P}_t^v and \mathbf{x}_t^u where, $u \in \mathcal{N}_t(v) \forall v \in [1, N_t]$
Output: a_t^v for $\forall v \in [1, N_t]$

- 1: Let $v = 1$ and $\forall j \in [1, N_t]$ set $\mathcal{A}_c^j = \emptyset$.
 - 2: **while** $v \leq N_t$ **do**
 - 3: Feed the TAG-GCN model with \mathbf{P}_t^v and \mathbf{x}_t^u , for all neighbor $u \in \mathcal{N}_t(v)$.
 - 4: Freeze all inputs of TAG-GCN except actions and predict the performance metrics $\hat{\alpha}_{t+1}^v(\cdot)$ and $\hat{\beta}_{t+1}^v(\cdot)$ for the input actions.
 - 5: **for** $|\mathcal{A}_c^v| \leq \nu$ **do**
 - 6: $x = \arg \max_{a \in \{\mathcal{A} - \mathcal{A}_c^v\}} -\sqrt{|1 - \hat{\beta}_{t+1}^v(a)|}$
 - 7: $\mathcal{A}_c^v = \mathcal{A}_c^v \cup \{x\}$
 - 8: **end for**
 - 9: $a_t^v = \arg \max_{a \in \mathcal{A}_c^v} \hat{\alpha}_{t+1}^v(a)$
 - 10: $v = v + 1$
 - 11: **end while**
 - 12: **return** a_t^v for $\forall v \in [1, N_t]$
-

$\gamma(\mathbf{y}_t^v, \mathbf{y}_t^u)$	$\mathbf{y}_t^v[0] > \mathbf{y}_t^u[0]$	$\mathbf{y}_t^v[0] \leq \mathbf{y}_t^u[0]$
$\mathbf{y}_t^v[1] \leq \mathbf{y}_t^u[1]$	2	1
$\mathbf{y}_t^v[1] > \mathbf{y}_t^u[1]$	3	4

Table 1: The relationship operator γ

Reward Model Accuracy Evaluation

Dataset Generation We use a simulator to modify Dataset-A with a random policy, in order to evaluate the prediction accuracy of their model. We randomly select the A2-threshold for each cell around the default action -100 dBm within the range of $[-105, -95]$ on each day. This approach provides a diverse dataset for training all models and allows for a fair comparison of their accuracy. On the other hand, for Dataset-B, which already has a reasonable amount of diversity in the handover parameter configuration, the raw dataset from the live network is used directly for training and evaluation.

Training Process and Metrics As samples are generated hourly, we aggregate them daily. To evaluate the model accuracy in predicting cell throughput and throughput ratio, we train the model with the generated pairs $\{(1, 2), \dots, (t-1, t)\}$ for $t = 9$ and 12 days for Dataset-A and B, respectively. At each day $t > 2$, data pairs $\{(1, 2), \dots, (t-2, t-1)\}$ are used as training and validation sets, and $(t-1, t)$ serves as testing set for evaluation across different models. We report the mean square error (MSE) to measure the reward model performance.

Comparison with Benchmark Models It is important to note that, the social impact of this work has not been addressed by ML approaches the same way as we propose. Due to the uniqueness of our problem, existing solutions for optimizing handover parameters are either not appropriate to solve it or there is no apparent way to directly adapt

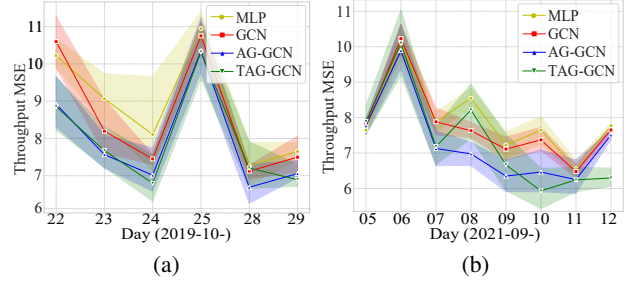


Figure 2: Achieved MSE of the throughput for test data of (a) Dataset-A and (b) Dataset-B for different methods

them to our problem. For instance, traditional handover optimization methods rely on designing fuzzy rules based on different measures of QoS in the network (Vasu et al. 2012), however, designing proper rules is complex and cannot handle the change in highly dynamic systems well. Instead, we hope to use a data-driven approach, among which the (deep) RL method gains the most attention (Cao et al. 2018; Wang et al. 2018). However, in this type of problems, the network provider only allows limited explorations of the parameter values (e.g., allows changing the A2 value once a day) to ensure the stability of the network. Thus, we only have limited days for exploring the best action, while RL models, usually need longer episodes to optimize the accumulated return.

In order to show the effectiveness of our proposed reward model, we compare it with alternative designs for the prediction model. It should be mentioned that all of these models are our contribution. The first model is MLP, where we only use the features of the center cells and ignore the neighboring cells' features. In GCN model, we follow the typical GCN formulation (Hamilton, Ying, and Leskovec 2017) and process the network as a homogeneous graph where the neighbor information is aggregated jointly without distinction. The AG-GCN model ignores the temporal dependencies of the data which we consider in TAG-GCN model. In Fig. 2, we compare the prediction accuracy of these models for throughput in Dataset-A and B. We observe on average the best accuracy for the test set is achieved by AG-GCN and TAG-GCN, with TAG-GCN performing marginally better on the average rank metric across the evaluation days, indicating that our neighbor aggregation and temporal features extraction have a considerable impact on the reward modeling for cellular networks. The same results also achieved for the throughput ratio model.

Overall Parameter Optimization Performance

The Action Recommendation Process In the following experiments we use the presented models to recommend the actions for Dataset-A. The actions in day 1, i.e., Oct. 17, have been set to the default value which is -100 dBm. Unless otherwise stated, the actions for the second day, i.e., Oct. 18, are initialized by a set of random values around the default action in the range of $[-105, -95]$. The model is trained iteratively on each day and used to recommend actions for the next day. The process is depicted in Fig. 3, where states of

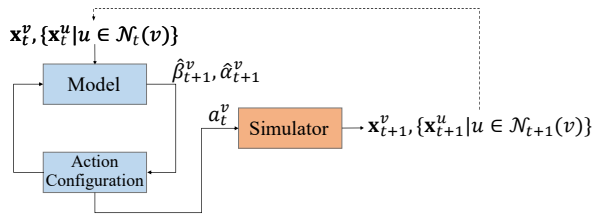


Figure 3: The process of action recommendation by the trained model and the simulator

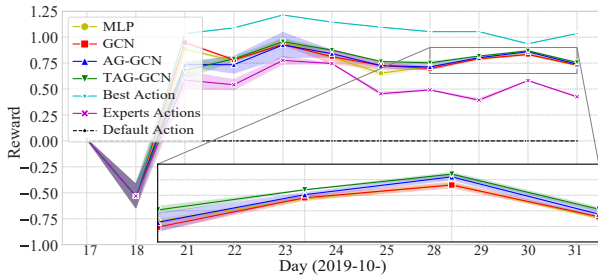


Figure 4: Performance comparison of different models along with optimal action curve initialized with random actions

the cells on day t are given to the trained model to predict performance metrics of the network on day $t + 1$ and the action a_t^v is adjusted for each cell based on the predictions. Finally, the network states and performance measurements for day $t + 1$ are computed according to the new selected action by the cellular network simulator and used for model training and action recommendation in the following day.

Baseline Performance Bounds In addition to the result achieved by the actions recommended by the models, we use three baseline performance bounds achieved by the *default A2-threshold*, the *expert rule*, and the *optimal actions* of the simulator. As stated before the default A2-threshold value is -100 dBm and this is used as the lower bound in the following experiments. The optimal actions in the simulator are obtained by brute-force search and it introduces the upper performance bound. The expert rule-based method is provided by experienced network operators which has a better performance than the default action. We hope to use our proposed learning based framework to further fill the gap with the reward achieved by optimal actions.

Results We plot the trajectory of the throughput difference to the default A2-threshold baseline (dash black line) in Fig. 4. We repeat all the experiments 20 times for all models, where each run uses the same set of random actions on the first action exploration day (Oct. 18) for all the models. We also show the performance achieved through the expert rule action recommendation, default action, and the optimal actions of the simulator (random actions are also used on Oct. 18 for the curve of the optimal action). TAG-GCN can achieve better average throughput in the final days which indicates the importance of our auto-grouping GCN design to tailor the heterogeneous property of the cellular networks.

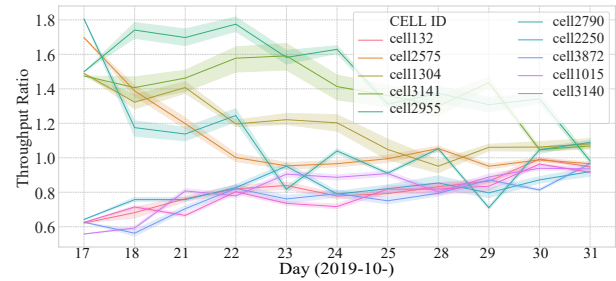


Figure 5: Trend of the throughput ratio for sample clusters

Besides, as expected, all the learning-based models can beat the expert rule algorithm which is highly dependent on human experience and is unable to recover from the performance degradation due to bad random initialization on the first day. Furthermore, to show the effectiveness of our proposed model in terms of load balancing and enhancing cluster throughput ratio, we illustrate the progress of this ratio achieved by TAG-GCN for some selected severely unbalanced cells in Fig.5. As it can be seen, the throughput ratio of the clusters forms a trajectory that converges to the ideal target value 1.

The experiments showed that the proposed ML-based solution improves network performance and optimizes handover compared to conventional methods. This reduces domain expert intervention, management costs, and improves maintenance efficiency, which can help provide reliable and high-quality network access to currently underserved areas. This could lead to new opportunities like remote education, remote working, health monitoring, and video streaming in those regions.

Conclusion

The paper proposes a novel approach called TAG-GCN to address the handover parameter configuration problem in cellular networks. This approach utilizes a reward prediction model to accurately estimate the performance metrics and investigate the impact of adjacent cells on the center cell of each cluster. The model also considers the temporal dependencies in the data to account for changing network dynamics. A multi-objective parameter configuration strategy is then proposed based on the reward model to optimize the performance metrics in each neighborhood. The simulations show that TAG-GCN outperforms existing methods and can lead to significant improvements in cellular network performance, which can have a positive social impact on sectors such as health and education.

References

3GPP TS36.331. 2016. Evolved Universal Terrestrial Radio Access (E-UTRA);Radio Resource Control (RRC); Protocol specification. <https://cdn.standards.iteh.ai/samples/48436/6b5fd0314d2944c89719e212cb6dba47/LTE-Evolved-Universal-Terrestrial-Radio-Access-E-UTRA-Radio-Resource-Control-RRC-Protocol-specification-3GPP->

- TS-36-331-version-13-0-0-Release-13-.pdf. Accessed: 2021-02-15.
- Agiwal, M.; Roy, A.; and Saxena, N. 2016. Next generation 5G wireless networks: A comprehensive survey. *IEEE Commun. Surveys Tuts.*, 18(3): 1617–1655.
- Attaran, M. 2021. The impact of 5G on the evolution of intelligent automation and industry digitization. *Journal of Ambient Intelligence and Humanized Computing*, 1–17.
- Bhat, P.; Nagata, S.; Campoy, L.; Berberana, I.; Derham, T.; Liu, G.; Shen, X.; Zong, P.; and Yang, J. 2012. LTE-advanced: an operator perspective. *IEEE Commun. Mag.*, 50(2): 104–114.
- Cao, G.; Lu, Z.; Wen, X.; Lei, T.; and Hu, Z. 2018. AIF: An artificial intelligence framework for smart wireless network management. *IEEE Commun. Lett.*, 22(2): 400–403.
- Chuai, J.; Chen, Z.; Liu, G.; Guo, X.; Wang, X.; Liu, X.; Zhu, C.; and Shen, F. 2019. A collaborative learning based approach for parameter configuration of cellular networks. In *Proc. IEEE Int. Conf. on Comput. Commun.*, 1396–1404.
- Dahlman, E.; Parkvall, S.; and Skold, J. 2013. *4G: LTE/LTE-advanced for mobile broadband*. Academic press.
- Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *Proc. Int. Conf. World Wide Web*, 417–426.
- Gohar, A.; and Nencioni, G. 2021. The role of 5G technologies in a smart city: The case for intelligent transportation system. *Sustainability*, 13(9): 5188.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep learning*. MIT press.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proc. Adv. Neural Inf. Proc. Systems*.
- Imran, A.; Zoha, A.; and Abu-Dayya, A. 2014. Challenges in 5G: how to empower SON with big data for enabling 5G. *IEEE Network*, 28(6): 27–33.
- Jiang, C.; Zhang, H.; Ren, Y.; Han, Z.; Chen, K.-C.; and Hanzo, L. 2016. Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Commun.*, 24(2): 98–105.
- Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learning Representations*.
- Mu, P.; Barco, R.; and Fortes, S. 2014. Conflict resolution between load balancing and handover optimization in LTE networks. *IEEE Commun. Lett.*, 18(10): 1795–1798.
- Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. In *Proc. Int. Conf. Learning Representations*.
- Rao, S. K.; and Prasad, R. 2018. Impact of 5G technologies on industry 4.0. *Wireless personal communications*, 100(1): 145–159.
- Riihijarvi, J.; and Mahonen, P. 2018. Machine learning for performance prediction in mobile cellular networks. *IEEE Comput. Intell. Mag.*, 13(1): 51–60.
- Siriwardhana, Y.; Gür, G.; Ylianttila, M.; and Liyanage, M. 2021. The role of 5G for digital healthcare against COVID-19 pandemic: Opportunities and challenges. *ICT Express*, 7(2): 244–252.
- statista. 2022. Number of smartphone subscriptions worldwide from 2016 to 2021, with forecasts from 2022 to 2027. <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. Accessed: 2022-07-22.
- Tabrizi, H.; Farhadi, G.; and Cioffi, J. 2012. Dynamic hand-off decision in heterogeneous wireless systems: Q-learning approach. In *Proc. IEEE Int. Conf. on Commun.*, 3217–3222.
- Tekinay, S.; and Jabbari, B. 1991. Handover and channel assignment in mobile cellular networks. *IEEE Commun. Mag.*, 29(11): 42–46.
- United Nations. 2015. United Nations Sustainable Development Goals. <https://sdgs.un.org/goals>. Accessed: 2022-07-22.
- Vasu, K.; Maheshwari, S.; Mahapatra, S.; and Kumar, C. S. 2012. QoS-aware fuzzy rule-based vertical handoff decision algorithm incorporating a new evaluation model for wireless heterogeneous networks. *EURASIP J. Wireless Commun. and Netw.*, 1–22.
- Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *Proc. ACM Int. Conf. Research and Development in Information Retrieval*, 165–174.
- Wang, Z.; Li, L.; Xu, Y.; Tian, H.; and Cui, S. 2018. Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. *IEEE Internet of Things Journal*, 5(6): 4296–4307.
- World Economic Forum. 2020. The Impact of 5G: Creating New Value across Industries and Society. https://www3.weforum.org/docs/WEF_The_Impact_of_5G_Report.pdf. Accessed: 2022-07-22.
- Ye, Q.; Rong, B.; Chen, Y.; Al-Shalash, M.; Caramanis, C.; and Andrews, J. G. 2013. User association for load balancing in heterogeneous cellular networks. *IEEE Trans. Wireless Commun.*, 12(6): 2706–2716.
- Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proc. ACM Int. Conf. Knowledge Discovery and Data Mining*, 974–983.
- Yu, C.; Liu, Y.; Yao, D.; Yang, L. T.; Jin, H.; Chen, H.; and Ding, Q. 2017. Modeling user activity patterns for next-place prediction. *IEEE Systems Journal*, 11(2): 1060–1071.
- Zhang, C.; Zhang, N.; Cao, W.; Tian, K.; and Yang, Z. 2020. An AI-based optimization of handover strategy in non-terrestrial networks. In *Proc. ITU Kaleidoscope: Industry-Driven Digital Transformation*, 1–6.
- Zhao, S.; Jiang, X.; Jacobson, G.; Jana, R.; Hsu, W.-L.; Rustamov, R.; Talasila, M.; Aftab, S. A.; Chen, Y.; and Borcea, C. 2020. Cellular network traffic prediction incorporating handover: A graph convolutional approach. In *Proc. IEEE Int. Conf. on Sensing, Commun., and Netw.*, 1–9.