INTEGRATED AND HETEROGENEOUS MOBILE EDGE CACHING (MEC) NETWORKS

Zohreh Hajiakhondi Meybodi

A thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montréal, Québec, Canada

December 2023

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Mrs. Zohreh Hajiakhondi Meybodi**
Entitled: **Integrated and Heterogeneous Mobile Edge Caching (MEC) Networks**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Andrea Schiffauerova ______	Chair
Dr. Shahrokh Valaee ______	External Examiner
Dr. Nizar Bouguila ______	External to Program
Dr. Hassan Rivaz ______	Examiner
Dr. Amir Asif ______	Examiner
Dr. Arash Mohammadi ______	Supervisor

Approved ______
Chair of Department or Graduate Program Director

______ 10/17/2023 ______ ______
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Integrated and Heterogeneous Mobile Edge Caching (MEC) Networks

Zohreh Hajiakhondi Meybodi, Ph.D.
Concordia University, 2024

The recent phenomenal growth of the global mobile data traffic, mainly caused by intelligent Internet of Things (IoTs), is the most significant challenge of wireless networks within the foreseeable future. In this context, Mobile Edge Caching (MEC) has been recognized as a promising solution to maintain low latency communication. This, in turn, improves the Quality of Service (QoS) by storing the most popular multimedia content close to the end-users. Despite extensive progress in MEC networks, however, there are still limitations that should be addressed. Through this Ph.D. thesis, first, we perform a literature review on recent works on MEC networks to identify challenges and potential opportunities for improvement. Then, by highlighting potential drawbacks of the reviewed works, we aim to not only enhance the cache-hit-ratio, which is the metric to quantify the users' QoS, but also to improve the quality of experience of caching nodes. In this regard, we design and implement a Deep Reinforcement Learning (DRL)-based connection scheduling framework [1] to minimize users' access delay by maintaining a trade-off between the energy consumption of Unmanned Aerial Vehicles (UAVs) and the occurrence of handovers. We also use D2D communication [2] to increase the network's capacity without adding any infrastructure. Another approach to effectively use the limited storage capacity of caching nodes is to increase the content diversity by employing the coded caching strategies in cluster-centric networks. Despite all the researches on the cluster-centric cellular networks, there is no framework to determine how different segments can be cached to increase the data availability in a UAV-aided cluster-centric cellular network. Moreover, to date, limited research has been performed on UAV-aided cellular networks to provide high QoS for users in both indoor and outdoor environments. Through this thesis research, we aim to address these gaps [3,4]. In addition, another goal of this thesis is to design real-time caching strategies [5–9] to predict the upcoming most popular content to improve the users' access delay. Last but not least,

capitalizing on recent advancements of indoor localization frameworks [10–14], we aim to develop a proactive caching strategy for an integrated indoor/outdoor MEC network.

Acknowledgments

First and foremost, I express my gratitude to Dr. Arash Mohammadi, my supervisor, whose exceptional guidance made this thesis possible. I also extend my thanks to the committee members—Dr. Shahrokh Valaee, Dr. Amir Asif, Dr. Nizar Bouguila, and Dr. Hassan Rivaz—for their insightful evaluation and feedback on this dissertation. Special thanks to Dr. Jamshid Abouei, my supervisor during my master's degree, whose encouragement propelled me to take this significant step, and Mr. Bastam, my high school teacher, who always believed in me. My heartfelt appreciation goes to my parents and siblings for their remarkable support, motivation, and empathy, and last but not least, my husband, Amir, who was always beside me on this journey and supported me with his love.

Contents

List of Figures

List of Tables

Chapter 1

Overview of the Thesis

Mobile Edge Caching (MEC) in Fifth Generation (5G) networks has been evolved as a promising solution to meet the phenomenal growth of the global mobile data traffic. The main idea behind MEC networks is to bring multimedia data to the edge of the network, i.e., close to User Equipments (UEs), to considerably reduce users' access delay while mitigating network congestion. More precisely, caching at the edge of the network provides the opportunity to store the most popular content at the storage of the caching nodes during the off-peak intervals. After requesting a content by a UE, this request is directly served by a neighboring caching node, having the requested content. In such scenarios, the cache-hit occurs; otherwise, it is known as a cache-miss and the requested content is sent from the content server to the caching node to serve the request. Although MEC network is positioned to provide seamless, secure, fast, and uninterrupted communication services, several critical challenges arise, especially in managing cache resources. *The primary research questions in MEC networks are where, how, and what to cache.* Below, several special characteristics of the MEC networks are introduced that might have significant influence on caching policies. By highlighting the existing challenges associated with MEC networks, the targeted research gaps and the main objectives of the thesis are then briefly described.

1.1 Main Challenges

In this section, an overall overview of the selected/targeted problems are provided by highlighting the main characteristics of the MEC networks. The key question of why one needs to do more research in the domain of MEC networks is then answered by emphasizing on existing and unsolved challenges. In particular, through this thesis research, we aim to partially or fully address the following identified challenges:

- **Caching Location:** One of the main challenges with MEC networks is determining where to cache the multimedia content, which refers to the selection of caching locations. One way to reduce the user experienced delay and the network's servicing cost is to employ small-cell terrestrial architectures (e.g., femtocells), called Femto Access Points (FAPs) [17], where femto base stations are equipped with the storage. In this case, network traffic can be handled by short range links through the nearest FAP. Another approach is to use Unmanned Aerial vehicles (UAVs) [20] as the aerial caching nodes to offload traffic via wireless backhaul, improve the network's coverage, and support a highly reliable and low-latency transmission. It should be noted that the wide transmission coverage of UAVs in comparison to FAPs, has brought several benefits, including the ability to handle the majority of users' requests and providing services for inaccessible areas. UAVs, however, suffer from limited battery life, particularly in situations where numerous active users in the network request contents. On the other hand, considering a dynamic MEC network, where users move consistently in the limited coverage area of FAPs, serving users' requests through FAPs leads to triggering frequent handovers. It is, therefore, critical to develop an efficient connection scheduling scheme to assign an appropriate caching node to UEs.

  Moreover, with the emphasis on the limited number of users that can be supported by each terrestrial/aerial caching node, another approach to increase the capacity of the network is to use direct communication between nearby users' devices, i.e., introducing the concept of Device-to-Device (D2D) communication. Consequently, these devices can be effectively employed as wireless caching nodes, and the number of caching nodes is inherently concentrated at those areas where the largest demands occur. Despite all the benefits that come with D2D communication, there is still a key challenge ahead. In particular, D2D communication leads to consuming the battery of mobile devices. Consequently, the amount of requests handled through the D2D communication should be managed.

- **Limited Storage:** To benefit from the MEC embedded in UAV-aided cellular networks, it is of significance importance to store the multimedia content at the edge of the network, before being requested. The limited storage of edge nodes, the large size of multimedia content, and the time varying behaviour of popularity of content, make it impossible to store all multimedia content at the edge nodes. Consequently, it is critical to efficiently and dynamically predict the popularity of content to identify/store the most upcoming requested ones.
- **UAVs' Signal Attenuation:** In MEC networks, incorporation of UAVs with terrestrial cellular infrastructure results in enhanced connectivity for outdoor users due to the high probability of establishing Line of Sight (LoS) links. Although the enhanced connectivity that comes by using UAVs will improve the Quality of Service (QoS) in outdoor areas, indoor penetration loss and deep shadow fading caused by building walls significantly attenuate the UAV's signals in indoor environments degrading the network's overall QoS. The UAV's limited battery life and its signal attenuation in indoor areas, therefore, make it inefficient to manage users' requests in indoor environments through UAVs. Consequently, it is essential to introduce an integrated MEC network to serve ground users positioned in both indoor and outdoor environments.
- **Small Coverage:** The transmission range of caching nodes (i.e., terrestrial and aerial caching nodes and Internet of Things (IoT) devices) is limited, therefore, caching nodes can only serve UEs located in their local transmission area. Given that users' preferences are influenced by their geographical location and contextual information, the optimal caching policy should take into account the UEs' location-based demands. Global Positioning System (GPS) is primarily utilized for tracking/monitoring UEs in outdoor environment, which is unable to offer reliable location information for indoor users. Therefore, it is of paramount importance to utilize an efficient indoor localization to identify UEs' locations and nearby caching nodes in indoor areas.

1.2 Targeted Research Gaps

Capitalizing on the identified challenges of MEC networks outlined in Section 1.1, in this section, we further identify and motivate key research gaps and questions that the thesis aims to tackle and address. In particular, the thesis focuses on the following key research gaps:

- **Network's Capacity:** One way to increase the network's capacity without adding any infrastructure is to use D2D communication. Due to the limited battery of mobile devices, UEs' limited transmission range, and security issues, however, some UEs may be unwilling to participate in D2D communication. Consequently, motivating users to initiate D2D communications has been an active research field for some times with several innovative and intelligent solutions being proposed regularly. Instead of employing a stand alone D2D-based network and dealing with its challenges, our target is to integrate D2D communication with aerial/terrestrial networks to reduce (manage) the number of requests handled by the D2D communication.

- **Heterogeneous Connections:** To take advantage of both terrestrial and aerial caching nodes, a heterogenous MEC network can be considered. Although potential benefits come by considering a heterogenous MEC network, several critical challenges arise, especially keeping a trade-off between multiple and conflicting objectives of different access points. More precisely, in an ultra-dense wireless network, where there are more than one possible caching node to handle user's request, UEs should be trained to determine the optimal caching node to minimize users' access delay by maintaining a trade-off between the energy consumption of UAVs and the occurrence of handovers. As will be discussed in the literature review section of the thesis, recent MEC scheduling research works mainly focused on one of these issues. This motivates us to address this lack of prior research studies on MEC scheduling to simultaneously satisfy concerns of UEs, UAVs, and FAPs.

- **Popularity Prediction:** Due to the limited storage of caching nodes, it is essential to identify/store the most popular content through the network to increase the number of requests that can be served directly by caching nodes. In the MEC networks, there are two types of caching strategies, i.e., reactive caching and proactive caching. Conventional reactive caching schemes identify the most popular content based on the underlying pattern of observed users' requests. A critical drawback of reactive caching is that popular content can only be identified after being requested. As a consequence, they are not robust to the dynamically changing behaviour of the content popularity. It is, therefore, crucial to use proactive caching, e.g., using Deep Neural Networks (DNN) models to predict the Content Popularity (CP) from the request patterns. In this context, popular content can be dynamically allocated in the storage of caching nodes before being requested. The thesis aims to further advance this emerging field.

- **Content Diversity:** Another way to effectively use the limited storage capacity of caching nodes is to increase the content diversity by employing coded caching strategies. In coded caching strategies, instead of storing a content completely into the storage of caching nodes, only specific segments of the content are stored. Early works on coded caching strategies considered *homogeneous* networks, where the same segments of the most popular multimedia content are stored in different caching nodes. To further increase the diversity of available content, it is beneficial to use the cluster-centric networks as a *heterogeneous* infrastructure, which is a combination of coded/uncoded caching schemes. Therefore, there should be an efficient placement strategy to determine: (i) How distinct segments of popular content should be distributed in different caching nodes, and; (ii) What is the best portion of the cache capacity, which is assigned to the coded/uncoded content placement in each caching node, to increase content diversity while decreasing users' access delay.

- **Integrated Indoor/Outdoor MEC Network:** The wide transmission range of UAVs and the high probability of establishing LoS links provide several advantages, including the ability to manage the majority of ground users' requests, which leads to improved coverage in outdoor environments. On the other hand, the transmitted signal by UAVs, propagating in residential areas, becomes weaker due to the penetration loss and shadow fading effects. Therefore, the requests that are handled by UAVs should be controlled. Another challenge of heterogenous MEC networks is the handover phenomenon, which can be frequently triggered by FAPs if the ground user moves rapidly and leaves the current position. To date, limited research has been performed on UAV-aided cellular networks to provide high QoS for ground users in both indoor and outdoor environments, which is a targeted gap of the thesis research.

- **Indoor Localization:** Several indoor localization strategies have been proposed to monitor/track users in indoor MEC networks, where the key challenge is susceptibility to the Non-Line-of-Sight (NLoS) error. In the presence of an obstacle between the anchor node and the mobile device, the received signal will be delayed, resulting in a positive bias and a significant degradation in the positioning accuracy. Therefore, NLoS mitigation/identification in indoor MEC networks is of paramount importance. Due to the complexity of analytical modelling of the multi-path and path- loss effects in indoor environments, the focus has shifted to data-driven approaches such as those based on supervised DNN. By considering the effects of multi-path and path-loss on the train dataset, therefore, one can eliminate the need for complex and precise analytical models. Despite all the benefits that come from using supervised models, there are several key challenges ahead. On the one hand, supervised models require labelled LoS/NLoS data, which is both costly and time-consuming limiting general applicability of supervised models within this context. On the other hand, even minor changes in the indoor environment would require updating the training dataset. Reinforcement Learning (RL) models, however, eliminate the necessity for labelling of channel conditions, therefore, allowing generalization while saving time and precious resources.

1.3 Research Objectives

Capitalizing on the need for addressing the existing challenges associated with MEC networks, through the Ph.D. thesis, we aim to achieve the following objectives:

(*Obj1*) **Indoor D2D-enabled MEC Network:** To take advantage of both D2D communication and terrestrial infrastructure while coping with their limitations, our goal is to introduce an integrated indoor D2D-based MEC network. With the emphasize on the UE's mobility, the main objective is to introduce a connection scheduling framework to reduce the UEs' energy consumption and avoid frequent terrestrial handovers. In this regard, the best caching node is selected by considering the Received Signal Strength Indicator (RSSI) value and the velocity of UEs as decision criteria to extend the time interval between two consecutive handovers and increase the battery life of devices.

(*Obj2*) **Autonomous Connection Scheduling:** In a heterogenous MEC networks, there are multiple conflicting objectives, such as user's access delay, UAVs' energy consumption and flying time, and terrestrial handover, that must be simultaneously satisfied. Conventionally, the focus of cache-enabled UAV-aided cellular networks was on deriving fixed mathematical/optimization models to meet predefined metrics. Such static models, however, are impractical due to the dynamic nature of wireless networks at hand such as unknown/varying number of active users, and adverse environmental conditions. To efficiently cope with the dynamic topology of heterogeneous networks and time-varying behaviour of ground users, our objective is to develop an efficient RL-based connection scheduling framework, where ground users are autonomously trained to determine the optimal caching node, i.e., UAV or FAP.

(*Obj3*) **Coded/Uncoded Content Placement:** In conventional femtocaching schemes, it is a common assumption that all caching nodes store the same most popular contents. This assumption is acceptable in static femtocaching models, in which users are stationary or move with a low velocity. With the focus on a dynamic femtocaching network, in which users can move randomly, storing distinct content in neighboring FAPs leads to increasing the number of requests served by caching nodes. Capitalizing on the Zipf law distribution for content popularity, the users' behaviour pattern for all content are not the same, i.e., while the majority of contents are not popular, some contents are requested moderately. In this context, the first objective of the thesis is to determine the best portion of the storage that should be allocated to the coded and uncoded contents for different content popularity profiles. The second objective is to determine how to distribute different segments of coded content within the network.

(*Obj4*) **Integrated Indoor/Outdoor MEC Network:** Capitalizing on the UAV's signal attenuation in indoor environments, our goal is to consider two different indoor and outdoor caching service scenarios. In this context and in line with advancements of 5G networks, we focus on the transmission scheme of indoor/outdoor users by defining a two phase clustering approach for FAPs' formation and UAVs' deployment. More precisely, the indoor area is covered by FAPs, equipped with extra storage and supported by the Coordinated Multi-Point (CoMP) technology. The outdoor area, however, is supported by coupled

UAVs and FAPs depending on the movement speed of ground users.

(*Obj5*) **Attention-based Popularity Prediction:** Incorporation of caching at the edge of the wireless networks significantly reduces the user's latency by bringing content closer to UEs. Due to the limited storage of edge nodes and the time variant users' preferences, it is essential to identify/store the most popular content dynamically. Time-variant CP contains temporal and spatial correlations of the targeted content. While spatial correlation reflects different users' preferences, depending on the geographical area and users' contextual information, the temporal correlation represents the variation of content popularity over time. Existing DNN models in this context suffer from long-term dependencies, computation complexity, and unsuitability for parallel computing. The goal of the thesis is to employ the Vision Transformer (ViT) architecture and design a real-time caching strategy, which to the best of our knowledge, is being studied for the first time.

(*Obj6*) **Self-Supervised Popularity Prediction:** Despite the various advantages of existing DNN techniques, a significant limitation of most of these models is their reliance on supervised learning. This means that these models require labeled samples to train, making it difficult to apply them to real-world problems where large amounts of labeled data may not be readily available. In cases where the dataset used for a study is unlabeled, manual labeling becomes necessary, which can be time-consuming and expensive. This need for manual labeling highlights one of the significant challenges of existing DNN-based popularity prediction frameworks, emphasizing the importance of developing highly accurate unsupervised/self-supervised learning models that can learn from unlabeled data, making them more adaptable to real-world situations.

(*Obj7*) **Mobility-Aware Indoor/Outdoor MEC Network:** Developing efficient indoor localization frameworks, to be integrated with the indoor/outdoor MEC network is my last objective. In the first step, we focused on development of DNN-based localization frameworks in different cutting-edge technologies such as Bluetooth Low Energy (BLE) and Ultra-WideBand (UWB). Since each of the indoor positioning technologies has its own benefits and limitations, we employ Angle of arrival (AoA) for BLE, and Time Difference of Arrival (TDoA) technique for UWB to meet their needs. My objective with the BLE-based AoA localization strategy was to employ a supervised learning model to monitor UE's movement in a dense indoor environment where the BLE's received signal is impacted by Rayleigh fading channel and elevation angle. Then, we introduced an RL-based UWB node selection framework to enhance the location accuracy, while maintaining a balance between the remaining battery life of UWB beacons.

1.4 Thesis Contributions

Below, the contributions of the thesis are briefly outlined:

- **Chapter 3 [1,5–8]: Autonomous Connection Scheduling and Uncoded Content Placement**
  - With focus on achieving *Obj2*, I developed the Convolutional Neural Network (CNN) and Q-Network-based Connection Scheduling (**CQN-CS**) framework [1]. The CQN-CS targets minimizing users' access delay by maintaining a trade-off between the energy consumption of UAVs and the occurrence of handovers. More specifically, the first objective of the proposed CQN-CS framework is allowing ground users to be autonomously trained to select an optimal caching node, i.e., UAV and/or FAP. Moreover, in contrary to existing research works, where only one criterion is optimized to improve the QoS of the network, the proposed CQN-CS framework simultaneously considers optimization of three metrics, which are the energy consumed by UAVs, the probability of FAP's handover, and the users' access delay. This contribution has already been achieved and published in IEEE Access [1].
  - Regarding *Obj5*, I developed a Vision Transformer-based Edge (**TEDGE**) caching framework as a real-time caching strategy with the application to the MEC networks is associated with *Obj5*. More precisely, the TEDGE caching scheme is a multi-label and time-series classification model with the aim of minimizing the difference between the actual Top-K popular content and the predicted ones. This contribution has already been achieved and published in IEEE International Conference on Communications (ICC) [5].

  - While TEDGE caching scheme provided high classification accuracy, the learning model was so complex. To reduce the number of parameters of the network, I proposed the parallel Vision Transformers with Cross Attention (**ViT-CAT**) model, consisting of two parallel ViT networks, one for collecting temporal correlation, and the other for capturing dependencies between different contents. Followed by a Cross Attention (CA) module as the Fusion Center (FC), the proposed ViT-CAT is capable of learning the mutual information between temporal and spatial correlations, as well, resulting in improving the classification accuracy, and decreasing the model's complexity about 8 times. This contribution has already been achieved and published in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) [6].
  - Targeting *Obj6*, I introduced a self-supervised learning algorithm called Contrastive learning Popularity (**CoPo**) prediction framework [7] to predict the dynamic content popularity in a MEC network. The framework utilizes the distinguishing aspect of the Contrastive Learning (CL) paradigm to recognize differences among input samples, including users' contextual information and is based on the Long Short Term Memory (LSTM) model to capture temporal information. This contribution has already been achieved and published in IEEE Digital Signal Processing [7]. Moreover, I extended this work by incorporating the CL framework with the Survival Analysis model to predict the time-to-popularity distribution of contents, which is accepted in IEEE Internet of Things Journal [8].

- **Chapter 4 [3, 4, 9]: Coded/Uncoded Content Placement**
  - Targeting *Obj3* and *Obj4* and to increase the content diversity in MEC networks, I proposed the Cluster-centric and Coded UAV-aided Femtocaching (**CCUF**) framework [3, 4]. In this work [4], I took advantage of the CoMP approach to further decrease the users' access delay by simultaneously serving multimedia content via multiple caching nodes. Moreover, I experimentally computed the number of coded contents to be stored in

  each caching node to increase the cache-hit-ratio, Signal-to-Interference-plus-Noise Ratio (SINR), and cache diversity and decrease the users' access delay and cache redundancy for different content popularity profiles. Finally, I introduced a novel transmission scheme to cover both indoor and outdoor environments. This contribution has already been achieved and published in IEEE Internet of Things Journal [4].

  - In accordance with the content placement strategy introduced in [4], I introduced a Transformer architecture to classify contents as either popular or unpopular in a coded/uncoded manner. It should be noted that most existing data-driven popularity prediction models are not suitable for the coded/uncoded content placement frameworks. On the one hand, in coded/uncoded content placement, in addition to classifying contents into two groups, i.e., popular and non-popular, the probability of content request is required to identify which content should be stored partially/completely, where this information is not provided by existing data-driven popularity prediction models. On the other hand, the assumption that users' preferences remain unchanged over a short horizon only works for content with a smooth request pattern. To tackle these challenges, I develop a Multiple-model (hybrid) Transformer-based Edge Caching (**MTEC**) framework with higher generalization ability, suitable for various types of content with different time-varying behavior, that can be adapted with coded/uncoded content placement frameworks.This contribution is under revision of IEEE Internet of Things Journal [9].

- **Chapter 5 [2, 10–14]: Indoor D2D-enabled MEC Network**

  - Targeting *Obj1*, I proposed a novel Mobility-Aware Femtocaching scheme based on Handover (**MAFH**) framework [2], where UEs are categorized based on their velocity into two groups; low speed and high speed clients. Since high speed users move and leave their current positions rapidly, D2D communication is just established between low speed users in order to reduce the energy consumption of UEs, while, all UEs, regardless of their speeds, allow to access their content through FAPs. Moreover, in order to reduce the number of unnecessary handovers, the appropriate target FAP for handover is chosen in such a way that the time interval between successive handover triggers will be extended, which consequently reduces the number of handovers during a connection. This contribution has already been achieved and published in IEEE Transactions on Vehicular Technology [2].

- Targeting *Obj7*, I proposed three indoor localization frameworks [10–12] with the application to the BLE technology. In [10], I proposed a set of signal processing and information fusion methods by integration of Nonlinear Least Square (NLS) curve fitting, Kalman Filter (KF), and Gaussian Filter (GF) to boost the accuracy rate of estimated angle. Moreover, in [11], I first highlighted the wireless signal model in BLE standard and formulate the transmitted signal, wireless channel model, and the signal received by Linear Antenna Array (LAA), to introduce a novel fusion processing technique to eliminate the destructive impact of the wireless channel on the received signal. Additionally, I developed an efficient CNN-based indoor localization framework [12] within the BLE-based settings. I considered an experimental indoor environment without presence of LoS links affected by Additive White Gaussian Noise (AWGN) with different Signal to Noise Ratios (SNRs) and Rayleigh fading channel. Moreover, by assuming a 3-D indoor environment, the destructive effect of the elevation angle of the incident signal is considered on the position estimation. These contributions have already been achieved, where they are published in the International Conference on Information Fusion, IEEE International Conference on Systems, Man, and Sybernetics (SMC), and IEEE International Conference on Acoustics, Speech and Signal Processing [12]
- Furthermore, I introduced an efficient node selection framework to enhance the location accuracy without using complex NLoS mitigation methods, while maintaining a balance between the remaining battery life of UWB beacons. Referred to as the Deep Q-Learning Energy-optimized LoS/NLoS (**DQLEL**) UWB node selection framework, the mobile user is autonomously trained to determine the optimal set of UWB beacons to be localized based on the 2-D TDoA framework. This contribution has

already been achieved and published IEEE Transactions on Signal Processing (TSP) [13].

- Finally, I proposed the Jump-start RL-based Uwb NOde selection (**JUNO**) framework [14], to accelerate the learning process of RL models. One of the most important challenges of existing RL models [13], is that the optimal policy should be learned by the interaction of the agent (mobile user) with the environment (i.e., via trial and error), without any prior information, especially when the model is just initialized. Consequently, it may take a long time for the RL model to reach the optimal policy. Another challenge is the generalization ability of the pre-trained RL model to be used in a new and different environment, where the density/location of obstacles is changing over the time/environment. To tackle these issues, we used Jump-Start RL (JSRL) model, where the agent use a guide-policy instead of a random one at the beginning of the learning process. Consequently, the learning process is accelerated and the RL generalization ability is highly improved. This contribution has already been achieved and published IEEE Global Communications Conference (GLOBECOM) [14]. The summary of contributions is presented in Table 1.1.

1.5 Organization of the Thesis

To provide the relevant context, the rest of the thesis is organized as follows.

- **Chapter 2** provides the literature review of MEC networks.
- In **Chapter 3**, I concentrate on the autonomous connection scheduling framework in cache-enabled UAV-aided cellular networks. Moreover, I introduce four popularity prediction models adopted with uncoded content placement frameworks.
- **Chapter 4** focuses on development of an integrated indoor/outdoor MEC network and coded/uncoded content placement, and designing of an attention-based popularity prediction.
- In **Chapter 5**, I focus on integrating the D2D communication in the indoor

Table 1.1: Summary of contributions.

Ref.	**Contribution**	**Indoor/Outdoor**	**Coded/Uncoded**	**Machine Learning**	**Caching Device**	**Localization**
[2]	Connection Scheduling	Indoor	Coded	-	D2D/FAP	-
[10]	Localization	Indoor	-	-	-	BLE/AoA
[11]	Localization	Indoor	-	-	-	BLE/AoA
[1]	Connection Scheduling	Outdoor	Uncoded	DRL	FAP/UAV	-
[12]	Localization	Indoor	-	CNN	-	BLE/AoA
[4]	Content Placement	Indoor/Outdoor	Coded/Uncoded	K-Means	FAP/UAV	-
[13]	Localization	Indoor	-	DRL	-	UWB/TDoA
[5]	Popularity Prediction	Outdoor	Uncoded	ViT	FAP/UAV	-
[14]	Localization	Indoor	-	RL	-	UWB/TDoA
[6]	Popularity Prediction	Outdoor	Uncoded	ViT	FAP/UAV	-
[7]	Popularity Prediction	Outdoor	Uncoded	CL	FAP/UAV	-
[8]	Popularity Prediction	Outdoor	Uncoded	CL	FAP/UAV	-
[9]	Popularity Prediction	Indoor/Outdoor	Coded/Uncoded	Trasnformer	FAP/UAV	-

MEC networks, as well as deploying indoor localization frameworks in presence of Rayleigh fading channel.

- **Chapter 6** concludes the thesis and future direction will be discussed.

1.6 Publications

Journal Publications

J5. **Z. Hajiakhondi-Meybodi**, A. Mohammadi, J. Abouei, and K. N. Plataniotis,“CLSA: Contrastive Learning-based Survival Analysis for Popularity Prediction in MEC Networks,” *Accepted in IEEE Internet of Things Journal*, August 2023.

J4. **Z. Hajiakhondi-Meybodi**, A. Mohammadi, M. Hou and K. N. Plataniotis, “DQLEL: Deep Q-Learning for Energy-Optimized LoS/NLoS UWB Node Selection”, *IEEE Transactions on Signal Processing*, vol. 70, pp. 2532-2547, 2022.

J3. **Z. Hajiakhondi-Meybodi**, A. Mohammadi, J. Abouei, M. Hou and K. N. Plataniotis, “Joint Transmission Scheme and Coded Content Placement in Cluster-Centric UAV-Aided Cellular Networks”, *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11098-11114, July 2022.

J2. **Z. Hajiakhondi-Meybodi**, A. Mohammadi and J. Abouei, “Deep Reinforcement Learning for Trustworthy and Time-Varying Connection Scheduling in a Coupled UAV-Based Femtocaching Architecture,” *IEEE Access*, vol. 9, pp. 32263-32281, Feb. 2021.

J1. **Z. HajiAkhondi-Meybodi**, J. Abouei, M. Jassemuddin and A. Mohammadi, “Mobility-Aware Femtocaching Algorithm in D2D Networks Based on Handover,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10188-10201, Sept. 2020.

Under Revision Journals

J1. **Z. Hajiakhondi-Meybodi**, A. Mohammadi, M. Hou, E. Rahimian, S. Heidarian, J. Abouei, and K. N. Plataniotis,“Multi-Content Time-Series Popularity

Prediction with Multiple-Model Transformers in MEC Networks," *Submitted to Ad hoc Networks*, Aug. 2023.

Conference Publications

C8. **Z. HajiAkhondi-Meybodi**, A. Mohammadi, J. Abouei, and K. N. Plataniotis, "CoPo: Self-supervised Contrastive Learning for Popularity Prediction in MEC Networks," *IEEE Digital Signal Processing*, June 2023.

C7. **Z. HajiAkhondi-Meybodi**, A. Mohammadi, M. Hou, J. Abouei, and K. N. Plataniotis, "ViT-Cat: Parallel Vision Transformers With Cross Attention Fusion for Popularity Prediction in MEC Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5.

C6. **Z. Hajiakhondi-Meybodi**, M. Hou and A. Mohammadi, "Content Placement in a Cluster-Centric Mobile Edge Caching Network," *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Shanghai, China, 2022, pp. 1-5.

C5. **Z. Hajiakhondi-Meybodi**, M. Hou and A. Mohammadi, "JUNO: Jump-Start Reinforcement Learning-based Node Selection for UWB Indoor Localization," *IEEE Global Communications Conference*, Rio de Janeiro, Brazil, 2022, pp. 6194-6199.

C4. **Z. Hajiakhondi Meybodi**, A. Mohammadi, E. Rahimian, S. Heidarian, J. Abouei and K. N. Plataniotis, "TEDGE-Caching: Transformer-based Edge Caching Towards 6G Networks," *IEEE International Conference on Communications*, Seoul, Korea, Republic of, 2022, pp. 613-618.

C3. **Z. HajiAkhondi-Meybodi**, M. Salimibeni, A. Mohammadi and K. N. Plataniotis, "Bluetooth Low Energy and CNN-Based Angle of Arrival Localization in Presence of Rayleigh Fading," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 7913-7917.

C2. **Z. Hajiakhondi-Meybodi**, M. Salimibeni, A. Mohammadi and K. N. Plataniotis, "Bluetooth Low Energy-based Angle of Arrival Estimation in Presence of Rayleigh Fading," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, 2020, pp. 3395-3400.

C1. **Z. Hajiakhondi-Meybodi**, M. Salimibeni, K. N. Plataniotis and A. Mohammadi, “Bluetooth Low Energy-based Angle of Arrival Estimation via Switch Antenna Array for Indoor Localization," *IEEE International Conference on Information Fusion (FUSION)*, Rustenburg, South Africa, 2020, pp. 1-6.

Other Research Publications

J3. M. Salimibeni, **Z. Hajiakhondi-Meybodi**, A. Mohammadi and Y. Wang, “TB-ICT: A Trustworthy Blockchain-Enabled System for Indoor Contact Tracing in Epidemic Control,” *IEEE Internet of Things Journal*, 2022.

J2. K. Azar, **Z. HajiAkhondi-Meybodi**, and F. Naderkhani, “Machine Learning-based Design of Maintenance Decision Support System for Fault Diagnostic and Prognostic,” *Reliability Engineering and System Safety*, 2022.

J1. A. Rasti-Meymandi, A. Madahian, J. Abouei, A. Mirvakili, **Z. HajiAkhondi-Meybodi**, A. Mohammadi, M. Uysal, “Design and Implementation of VLC-based Smart Barrier Gate Systems,” *International Journal of Electronics & Communications*, 2021.

C5. B. Sarhadi, J. Abouei, **Z. HajiAkhondi-Meybodi**, and A. Mohammadi, “Energy Efficient Clusterbased Image Compressive in Wireless Multimedia Sensor Networks,” *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, July 2021.

C4. A. Rasti-Meymandi, J. Abouei, **Z. HajiAkhondi-Meybodi**, A. Mohammadi, and, A. Asif, “Fast machine learning-based signal classification in energy constrained CRN: FPGA design and implementation,” *IEEE International Conference on Autonomous Systems (ICAS)*, Feb. 2021.

C3. M. S. Beni, **Z. HajiAkhondi-Meybodi**, A. Mohammadi, et al., “IoT-TD: IoT Dataset for Multiple Model BLE-based Indoor Localization/Tracking,” *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, 2021.

C2. M. Atashi, M. S. Beni, P. Malekzadeh, **Z. HajiAkhondi-Meybodi**, K. N. Plataniotis, and A. Mohammadi, “Orientation-Matched Multiple Modeling for RSSI-based Indoor Localization via BLE Sensors,” *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1702-1706.

C1. M. Asadi, J. Abouei, **Z. HajiAkhondi-Meybodi**, M. Mazidi, and A. Mohammadi, “FDIA Detection through an Adaptive Multi-Level Features Classification in Smart Grids,” *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, Oct. 2020, pp. 1828-1833.

Chapter 2

Literature Review

In this chapter, the focus is on providing an overview of MEC networks from the aspect of the caching location, connection scheduling, content placement, and popularity prediction. In addition, an overview of different indoor localization schemes is provided to find an efficient solution for indoor mobility-aware MEC networks.

2.1 Connection Scheduling

Recently, several promising approaches have been developed to store the most popular multimedia contents in local caching nodes, including femtocaching architectures [17], Device-to-Device (D2D) communications [18], UAV-based frameworks [19], and a combination of above schemes [20]. In this regard, using UAVs over wireless networks helps FAPs to offload traffic via wireless backhaul, improve the network's coverage, and support a highly reliable and low-latency transmission. With the emphasis on the features of high mobility and low-cost manufacturing, the satisfactory rate of users will increase through the utilization of UAVs as additional caching nodes for providing services to ground users. Consequently, there has been recent widespread attention to UAVs due to their impressive potentials in supporting a wide range of commercial and industrial applications. UAVs provide an adaptive platform, that can be altered by time-varying states of the environment and the need of ground users. Recent developments on stand-alone UAV-based networks have brought several benefits, including but not limited to wide coverage and low-cost services. Therefore, several UAV-based small cell networks were introduced, such as the UAV clustering scheme [22], where UAVs play the role of caching nodes to serve all terrestrial users in the network. Reference [23], for instance, presented a wireless network architecture that employed cache-enabled UAVs with the goal of achieving considerable improvements in the users' Quality of Experience (QoE). A UAV-based machine learning algorithm was employed in [24] to predict the distribution of video content requested based on Echo State Networks.

The wide transmission range of UAVs and the high probability of establishing LoS links provide several advantages, including the ability to manage the majority of ground users' requests, which leads to improved coverage in outdoor environments. Due to the limited battery life of UAVs, however, requests that are handled by UAVs should be controlled. Another challenge is the handover phenomenon, which can be frequently triggered by FAPs if the ground user moves rapidly and leaves the current position [25]. To overcome these issues, the UAV-aided cellular network is considered as a combination of UAVs and FAPs to mitigate the traffic load on either of the two. In this case, the overload on the backhaul link and the energy consumption of UAVs reduce significantly. Although potential benefits come by coupling UAVs with FAPs for the development of advanced femtocaching strategies, several critical challenges arise, especially keeping a trade-off between the energy consumption of UAVs and handover, occurring between FAPs. It is, therefore, critical to develop an efficient connection scheduling scheme to assign an appropriate caching node to ground users.

Generally speaking, the overall objective of connection scheduling in UAV-aided cellular networks is to improve the QoS of ground users, UAVs, and FAPs. In this regard, there are several conflicting objectives such as users' access delay, energy consumption of UAVs, and handover phenomena occurred between FAPs. Recent connection scheduling research works, however, mainly focused on one of these issues. For instance, existing solutions [26–28] focused on the energy consumption of UAVs. In Reference [26], a neural-blockchain-based UAV-caching approach is designed to provide a high reliability communication in terms of improvement in the energy consumption of UAVs, the maximum failure rate, the probability of connectivity, and survivability. Authors in [27] proposed a closed-form model for the energy consumption of rotary-wing UAVs with the aim of optimizing UAVs' trajectories and required communication times allocated to handle user requests. In [28], a fixed altitude for UAV's fly is assumed and an Air-to-Ground (A2G) communication scheduling scheme is proposed to optimize the trajectory, transmit power, and speed of UAVs, with the focus on decreasing the UAV propulsion energy consumption. In summary, the focus of these research works is on optimizing the energy consumption of UAVs, however, without addressing challenges associated with additionally introduced delays in the system that users experience when their requests are served through UAVs. On the other hand, Athukoralage *et al.* [29] considered a connection scheduling framework, where ground users are supported by UAV or WiFi APs. In this work, the users' link quality is utilized to balance the load between UAVs and WiFi APs. Zhu *et al.* [30] proposed a game theory-based connection scheduling scheme, where the probability of packet collision is used to select the optimal AP among all possible UAVs and Base Stations (BSs). To the best of our knowledge, there is no framework concerning the problem of autonomous connection scheduling between FAPs and UAVs. Furthermore, to date, limited research has been performed on UAV-aided cellular networks to provide high QoS for ground users in both indoor and outdoor environments.

2.2 D2D-based Caching Frameworks

Recent developments in modern users' devices provide a considerable storage capacity in smart phones. Consequently, these devices can be effectively employed as wireless caching nodes introducing the concept of D2D communication [32]. One of the particularly highlighted advantages of D2D communication is that the number of caching nodes is inherently concentrated at those areas where the largest demands occur. In [33], the authors introduced the Multi-Agent

Multi-Armed Bandit (MAMAB) scheme, in which MUs collaboratively cache and share contents to alleviate the burden of main server. Authors in [34] considered both self-caching and D2D communication in a clustering network to attain a sufficient throughput and energy efficiency, even though these two performance metrics are conflicting design objectives. A trade-off is, therefore, needed to achieve a balanced performance. Despite all the benefits that come with D2D communication, motivating users to establish D2D communications has been an active research field for some times now with several innovative and intelligent solutions (e.g., [35–37]) being proposed regularly. For example, the authors in [35] introduced a social-monetary-aware incentive model focused on the social links, transmission delay costs, and content fees to motivate all mobile users (transmitter and receiver) to connect via D2D communication. As a coalition forming method, the authors in [36] proposed an optimization problem to stimulate the requirement for D2D communications through a simple model of pricing in cellular networks. In this situation, consumer devices are encouraged to collaborate by taking into account both the cost and the transmission delay in decreasing their expenses. The authors in [37] presented a two-stage non-cooperative game to analytically model a discount interference pricing scheme for D2D communication.

Most of the previous caching proposals for wireless networks assumed fixed locations for users' devices (e.g., [38]). Since user mobility is an intrinsic feature of the next generation of wireless networks, in recent years, the focus has shifted to mobility-aware caching methods to achieve the best possible performance [39]. Note that the user mobility offers both advantages and disadvantages in femtocaching networks. Mobility provides more communication opportunities for mobile users to contact with more users and access more segments of a requested content. Therefore, the probability of failed file delivery in femtocaching networks will decrease. However, mobility of both users and caching users' devices over time, engenders fluid network topology. Furthermore, accessing to the history of users' requests to predict the most popular files is not useful [40]. There are some preliminary studies considering users' mobility in wireless networks based on the D2D communication. The authors in [41] showed the impact of mobility on caching strategies with regard to providing storage on FAPs and mobile users. They proposed a mobility-aware cache placement method to achieve a high cache hit ratio and minimize the energy consumption of clients. The authors in [39] introduced the Mobility-Aware Femtocaching scheme based on Markov Chain (MAFMC) pattern as an optimization problem to model user motions and utilizing network coding to reduce the MBS load. However, similar to most literature, the handover phenomenon in D2D-aided cellular networks, occurred during the movement of users, is ignored [39, 41].

2.3 Coded Content Placement

The main objective of UAV-aided cellular networks is to bring multimedia data closer to ground users and simultaneously improve users' QoS and the network's QoE. If the requested content can be found in the storage of one of the available caching nodes, this request would be served directly and cache-hit occurs; otherwise, it is known as a cache-miss. Due to the large size of multimedia contents, however, it is not feasible to store all contents in the storage of caching nodes. To increase content diversity, coded caching strategies [42, 43] have received remarkable attention lately. In coded caching strategies, only specific segments of the most popular multimedia contents are stored in the caching nodes. Early works on coded femtocaching such as Reference [44], however, considered homogeneous networks, where the same segments of the most popular multimedia contents are stored in different caching nodes. Non cluster-centric approaches [44], generally, ignore potentials of using the Joint Transmission (JT) scheme, which if utilized can both improve the QoS of edge-users and increase diversity of available contents for ground users. Therefore, the main focus of recent researchers has been shifted to the cluster-centric networks [45] as a heterogeneous infrastructure, where distinct segments are stored in neighboring caching nodes.

Cluster-centric cellular networks provide several benefits, such as increased content/cache diversity, which in turn leads to an increase in the number of requests managed by the caching nodes. However, this comes with the cost of experiencing inter-cell interference, especially for cell-edge users. To mitigate the inter-cell interference and improve the throughput of the cell-edge users, content caching in CoMP-integrated networks [45–50] has been studied in recent years. For instance, Chen *et al.* [45] developed two transmission schemes, namely JT and Parallel Transmission (PT), which are selected based on the popularity of the requested content. Alternatively, Lin *et al.* [48] proposed a cluster-centric cellular network applying the CoMP technique based on the users' link quality, where cell-core and cell-edge users are served through Single Transmission (ST) and JT, respectively. Despite all the researches on the cluster-centric cellular networks, there is no framework to determine how different segments can be cached to increase the data availability in a UAV-aided cluster-centric cellular network to increase content diversity.

2.4 Popularity Prediction

In the MEC networks, there are two types of caching strategies, i.e., Reactive caching and proactive caching. Conventional reactive caching schemes [51], such as First-In-First-Out (FIFO), Least Recently Used (LRU), and Least Frequently Used (LFU) frameworks, identify the most popular content based on the underlying pattern of observed users' requests. A critical drawback of reactive caching is that popular content can only be identified after being requested. As a consequence, they are not robust to the dynamically changing behavior of the content popularity. Therefore, the main focus of recent researches has been shifted to use proactive caching, e.g., using DNN models to predict the CP from the request patterns. In this context, popular content can be dynamically allocated in the storage of hgNBs before being requested. The paper aims to further advance this emerging field.

Generally speaking, both temporal and spatial correlations exist within the time-variant request pattern of multimedia content. While spatial correlation reflects different users' preferences, depending on the geographical location and users' contextual information, the temporal correlation represents the variation of content popularity over time. In this context, several DNN models [52–58,60,130] are introduced to capture the temporal and/or spatial features of user preferences in proactive caching schemes. For instance, Yu *et al.* [61] used an auto-encoder model to predict users' preferences in the future by learning the latent representation of raw data in an unsupervised fashion. Auto-encoder models, however, suffer from training complexity. Tsai *et al.* [62] used Convolutional Neural Network (CNN) for predicting users' interests based on sentence analysis. Ndikumana *et al.* [63] introduced a DNN-based caching framework, compromising of Multi-Layer Perceptron (MLP) and CNN models, where contextual information such as age, emotion, and gender are utilized for making caching decisions. Although CNN-based proactive caching schemes have local spatial feature awareness, they are inefficient for extracting temporal features from the patterns of sequential requests. Furthermore, such models require multi-source input such as regional information, and contextual information of users to improve the cache performance. Therefore, they need an efficient data pre-processing model to extract this information.

To deal with the time-varying behavior of request patterns, Recurrent Neural Networks (RNNs), such as Long Short Term Memory (LSTM) [55,64], are introduced to use historical information of the content. To extract both spatial and temporal features of CP data, Ale *et al.* [53] used a combination of LSTM and CNN models. LSTM-based caching frameworks, however, suffer from long-term dependencies, computation complexity, and unsuitability for parallel computing. To address challenges associated with RNN architectures, the Transformer neural network [65] has been designed to handle sequential input data, which is purely reliant on attention mechanisms with no recurrence or convolutions. One of the most important advantages of Transformers over RNN models is the attention mechanism, which eliminates the need to analyze data in the same order. Consequently, Transformers have higher parallelization capabilities than RNNs, implying reduced training time.

2.5 BLE-based Indoor Localization

Recent developments and advancements in the Internet of Things (IoT), low power wireless networks, and processing methodologies have resulted in the emergence of several different and innovative indoor localization technologies, including Infrared (IR) systems [66], Ultrasonic (US) systems [67], and Optical-based frameworks [68]. These systems share several common underlying properties, such as being sensitive to multipath effects, high costs and complexity and in some cases, and not being readily available [69]. Consequently, the main focus of researchers has been shifted to Radio Frequency (RF) indoor localization technologies, including Radio-Frequency IDentification (RFID) [70], ZigBee [71], WiFi [72], BLE [73], and UWB [74]. There are several notable factors that should be considered in developing an indoor localization system, such as cost, accuracy, robustness, scalability, power requirements, reliability, and coverage. Over the last few decades, there has been a significant surge of interest for BLE-based technologies, as one of the most reliable RF-based localization frameworks due to its availability (e.g., BLE is available on most modern smart devices), low power consumption, and low cost [75].

BLE is a commonly used low-power wireless protocol for IoT applications. Bluetooth beacons are small radio transmitters that send signals within upto 100 m radius. Bluetooth beacons, supported by many systems, are cost effective and are able to accurately determine the location of user devices upto few meters. There are numerous indoor location schemes that works on different aspects of BLE received signal. For instance, Received Signal Strength (RSS)-based methods [76] determine the location of user devices based on the strength of the signal, received by BLE beacons, while AoA [77] and Time of Arrival (ToA) [78] schemes evaluate the angle and the time of the incident signal, respectively.

AoA-based localization, as an active research field for several decades, is a non-linear triangulation approach to measure the position of mobile agents based on the direction of the incident radio frequency signal, received by an antenna array such as Linear Antenna Array (LAA) [78]. Subspace-based angle estimation algorithms [79], such as MUltiple Signal CLassification (MUSIC) and its extensions, are among the early research efforts for AoA estimation. Despite the benefits that can be obtained by using subspace-based angle estimation techniques, such localization methods suffer from some drawbacks. A key limitation is the unreliability of the subspace-based algorithms in the presence of the multi-path effect, which is an unavoidable factor in indoor environments [80]. Multi-path fading channel in indoor environments is commonly modelled by statistical models mainly Rayleigh [81] and Rician [82]. By assuming that there is a strong LoS path between the transmitter and the receiver, there are a wide range of approaches to address the multi-path propagation, including channel classification [83], Kalman filter-based techniques [84], and subsample interpolation methods [85]. Presence of different objects within indoor environments, however, leads to receiving the reflected, refracted, diffracted, and scattered versions of the transmitted wireless signal. Consequently, the assumption of existing a strong LoS path is not practical in most indoor localization scenarios. On the other hand, due to the complexity of analytical modelling of the multi-path and path-loss effects in indoor environments, the focus has shifted to data-driven approaches such as those based on DNNs [86]. Therefore, by considering the effects of multi-path and path-loss on the train dataset, one can eliminate the need for complex and precise analytical models. Capitalizing on these advantages, for instance Reference [87] introduced a CNN-based localization approach for the 2-D AoA estimation in the presence of noise. Authors in [88] investigated the effect of noise in a 3-D environment on the angle estimated by employing DNNs. Authors in [89, 90] proposed a DNN-based localization framework, where the input of the DNN is the Channel Impulse Response (CSI)-AoA images. CSI, however, is prone to the noise, shadowing, and small scale fading, leading to a considerable localization error.

Despite all the research conducted in this field, research on data driven BLE-based AoA localization is very limited [88] as only recently direction-finding feature is introduced to the BLE specification. In particular, the challenge of modelling the wireless channel as a combination of Rayleigh fading and noise without presence of the LoS link between the transmitter and the receiver in a 3-D indoor environment has not yet been considered. In such real indoor environments, the mobile user and BLE beacons are not always located along the same line, which in turn leads to elevation angle. Although the azimuth angle of the incident signal is utilized for location estimation, the destructive effect of elevation angle should be considered.

2.6 UWB-based Indoor Localization

UWB technology has been emerged as a solution to meet the phenomenal growth of the need for localizing users in indoor environments [93]. The use of a wide radio spectrum in UWB technologies enables individual multi-path components of the received signal to be efficiently resolved, resulting in high accuracy indoor positioning [94]. To monitor/track users in indoor environment, several localization techniques have been proposed such as RSSI, AoA, and TDoA [95, 96], among which time-based solutions [95–97] are the most efficient ones. In such scenarios, the time of the received signal from a set of available UWB beacons is required, which can only be estimated accurately if the first arrival path has been properly identified. In this context, one key challenge is susceptibility to the NLoS error. In presence of an obstacle between the UWB beacon and the mobile device, the time of the received signal will be delayed, resulting in a positive bias and a significant degradation in the positioning accuracy. Therefore, NLoS mitigation/identification in UWB-based indoor localization is of paramount importance. Conventional indoor localization frameworks reduced such location error via parametric solutions [98], the accuracy of which is dependent on implementation of complex pre-processing techniques adding considerable latency. Furthermore, using a large number of UWB beacons for localizing users is inefficient from the energy consumption perspective. In this regard, Dai *et al.* [99] analytically proved that tracking users' locations through a subset of active beacons offers several benefits, including mitigating the energy consumption of beacons, and improving the location accuracy. Consequently, the main focus of recent researches [100–103] has been shifted to use anchor node selection to achieve the best localization performance in terms of location accuracy and resource management. The paper aims to further advance this emerging field.

Anchor node selection in the context of indoor localization is utilized to improve the network's performance by setting a set of criteria for selecting a subset of beacons with the highest utilities. One of the most important criteria in indoor localization is to mitigate the location error, caused by NLoS connections. Towards this goal, LoS connections will be selected for monitoring/tracking users' locations instead of extracting location information from all available beacons. Within the Wireless Sensor networks (WSNs) context, Zhu *et al.* [104] used the Link Condition Indicator (LCI) as a metric to illustrate the link quality (i.e., LoS/NLoS conditions) of anchor nodes. Dai *et al.* [99] proposed a near optimal solution for identifying LoS/NLoS connections, which is used for power allocation and anchor node selection in Wireless Network Localization (WNL). Zhang *et al.* [105] introduced the topological unit, based on the topological relationship between nodes, to mitigate the location error caused by NLoS connections in an indoor environment.

With the application to the UWB-based indoor localization, Wang *et al.* [106] calculated the CIR from all beacons to determine two LoS connections out of all beacons to be involved in the localization. Albaidhani *et al.* [101] proposed an UWB node selection based on the Mean Squared Error (MSE) metric, where UWB beacons are clustered into different groups. Then, the group with the lowest MSE is selected to be used for localization by using the Weighted Least Square (WLS) method. Albaidhani *et al.* [103] introduced another evaluation metric for selecting the best set of UWB beacons, named Geometric Dilution of Precision (GDOP), and showed the superiority of the GDOP in comparison to the MSE metric in terms of the location accuracy. Despite all the benefits that come from using existing anchor node selection frameworks, there are still critical challenges ahead. On the one hand, energy consumption efficiency is compromised in scenarios where all the beacons require to transmit/receive signals for LoS/NLoS identification. On the other hand, analytical anchor node selection frameworks developed based on fixed mathematical models fail to cope with the dynamic nature of indoor environments, such as unknown/varying adverse environmental conditions.

Chapter 3

Autonomous Connection Scheduling and Uncoded Content Placement

In this Chapter, an ultra-dense MEC network consisting of FAPs and UAVs is considered, where there are more than one possible caching node to handle user's request. To meet the dynamic nature of Heterogeneous Networks (HetNets) such as unknown/varying number of active users, and adverse environmental conditions, the focus of recent research works have been shifted to the RL approaches instead of using fixed mathematical models. Therefore, ground users can be autonomously trained to determine the optimal caching node, i.e., UAV or FAP. Generally speaking, the overall objective of connection scheduling in HetNets is to improve the QoS of ground users, UAVs, and FAPs. In this regard, there are several conflicting objectives such as users' access delay, energy consumption of UAVs, and handover phenomena occurred between FAPs. Recent HetNet scheduling research works, however, mainly focused on one of these issues. The main motivation behind this work is to address this lack of prior research studies on HetNet scheduling to simultaneously satisfy concerns of ground users, UAVs, and FAPs. To achieve these objectives, we formulate a multi-objective optimization problem and propose the CNN and Q-Network-based Connection Scheduling (CQN-CS) framework. More specifically, to solve the constructed multi-objective connection scheduling problem, a deep Q-Network model is developed as an efficient RL approach to train ground users to handle their requests in an optimal and trustworthy fashion within the coupled UAV-based femtocaching network. The effectiveness of the proposed CQN-CS framework is evaluated in terms of the cache-hit ratio, user's access delay, energy consumption of UAVs, and FAP's handover. Based on the simulation results, the proposed CQN-CS framework illustrates significant performance improvements in companion to Q-learning and Deep Q-Network (DQN) schemes across all the aforementioned aspects.

Although massive connectivity enabled by the autonomous connection scheduling in MEC networks will significantly increase the quality of communications, there are several key challenges ahead. The limited storage of edge nodes, the large size of multimedia content, and the time-variant users' preferences make it critical to efficiently and dynamically predict the popularity of content to store the most upcoming requested ones before being requested. Recent advancements in DNNs have drawn much research attention to predict the content popularity in proactive caching schemes. Existing DNN models in this context, however, suffer from long-term dependencies, computational complexity, and unsuitability for parallel computing. To tackle these challenges, we propose several attention-based popularity prediction methods suitable for uncoded content placement. Finally, we introduce a self-supervised popularity prediction model to eliminate the need for manually lablelling available datasets.

3.1 Autonomous Connection Scheduling

The main novelty of this work is the design of an autonomous and decentralized scheduling approach via simultaneous incorporation of three key objectives, i.e., users' access delay, energy consumption of UAVs, and handover phenomena. In other words, the proposed CQN-CS framework allows ground users to autonomously determine (via the RL-based component) an optimal caching node to handle their requests without reliance on any central processing unit.

Table 3.1: List of Notations.

Notation	**Description**	**Notation**	**Description**
N_g	Number of ground users	$\mathbf{L}_j(t)$	Location of ground user GU_j at time slot t
N_f	Number of FAPs	$\boldsymbol{v}_j(t)$	Velocity of ground user GU_j at time slot t
N_u	Number of UAVs	h_k	Altitude of UAV u_k
N_c	Number of FAPs covered by a UAV	$d_{c,k}(t)$	Euclidean distance between the cloud server and UAV u_k
$\mathcal{C}$	Number of contents	$d_{k,j}(t)$	Euclidean distance between UAV u_k and ground user GU_j
R_f	Transmission range of FAPs	$\mathcal{L}_0$	Path loss in reference distance d_0
R_u	Transmission range of UAVs	$\mathcal{L}_{k,j}^{(LoS)}(t)$	LoS path loss from UAV u_k to ground user GU_j
S_f	Storage capacity of FAPs	$\overline{\mathcal{L}}_{k,j}(t)$	Average path loss from UAV u_k to ground user GU_j
S_u	Storage capacity of UAVs	α, β	Path loss exponent of LoS and NLoS links between server and u_k
L_c	Size of file c_l	$\eta^{(LoS)}$	Path loss exponent of LoS
γ	Popularity skewness c_l	$\chi_{\sigma}^{(LoS)}$	Shadowing effect of LoS link
p_l	Probability of requesting content c_l	τ_f, τ_p	Flyby and the pause time of UAV u_k
$p_k^{(h)}(t)$	Probability of cache-hit at time slot t	$\mathcal{U}_j^{(f)}$	Set of accessible FAPs for ground user GU_j
$p_k^{(m)}(t)$	Probability of cache-miss at time slot t	$\mathcal{U}_j$	Set of FAPs f_i and UAV u_k, associated with ground user GU_j
$p_{k,j}^{(LoS)}(t)$	Probability of LoS link u_k and GU_j	$E_k^{(LoS)}(t)$	Energy consumption of UAV u_k over LoS link
$\mathcal{D}_u(t)$	User's access delay through UAVs	$P_T(t)$, $P_R(t)$	Transmission and reception powers of 1 Mb file sharing
$\mathcal{D}_f(t)$	User's access delay through FAPs	$P_j(t)$	Received power of UAV u_k at ground user GU_j
P_k	Constant transmit power of UAV u_k	P_{th}	Minimum strength signal detected by ground users
N_0	Noise power	$RSSI_{i,j}(t)$	Received signal strength by ground user GU_j from FAP f_i
$I_k(t, \mathbf{u}_{-k})$	Interference of UAV-user links for u_k	$\mathcal{HO}_i(t)$	Handover parameter

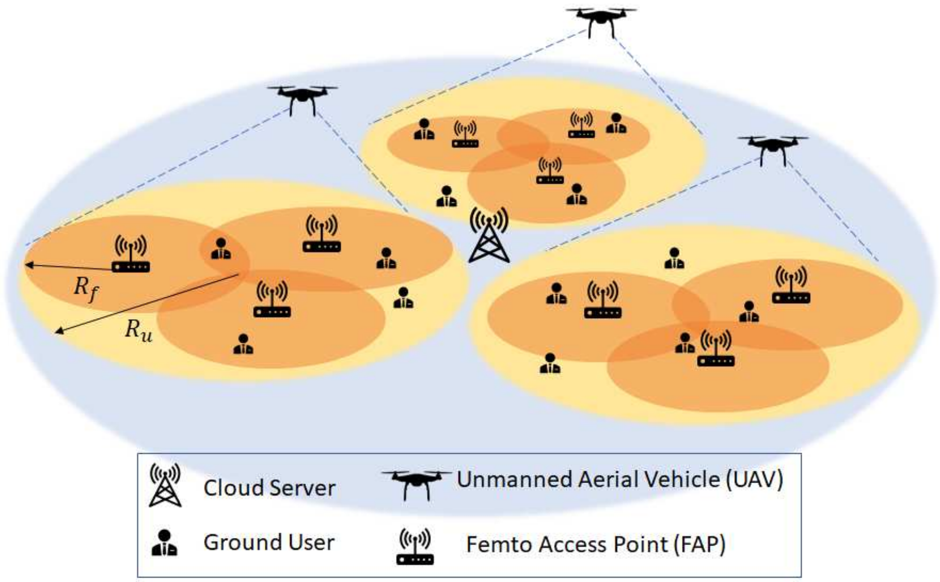


Figure 3.1: A typical structure of the proposed UAV-based femtocaching model.

3.1.1 System Model and Problem Description

In this work, we consider a UAV-based femtocaching network consisting of a set of ground mobile users, denoted by GU_j, for $(1 \leq j \leq N_g)$, N_f number of FAPs, denoted by f_i, for $(1 \leq i \leq N_f)$, N_u number of UAVs, as the flying caching nodes, denoted by u_k, for $(1 \leq k \leq N_u)$, and a cloud server. As depicted in Fig. 3.1, the transmission range of FAPs, denoted by R_f, is much more limited than that of UAVs, denoted by R_u, where a UAV covers N_c number of FAPs in a neighbourhood. Without loss of generality, it is assumed that similar to FAPs that have the same storage of size S_f, UAVs are equipped with the cache storage with an equal size of S_u.

In this model, ground users request an equal-sized video file c_l, for $(1 \leq l \leq \mathcal{C})$, with the probability of p_l, where p_l is calculated based on the Zipf-like distribution as follows [107]

$$p_l = \frac{l^{-\gamma}}{\sum\limits_{r=1}^{\mathcal{C}} r^{-\gamma}}, \tag{1}$$

where $\mathcal{C}$ denotes the total number of video contents in the network, stored at a cloud-based content server, known as the main server in the network, and γ represents the popularity skewness. In a UAV-based femtocaching network, if a requested content exists in the caching nodes including FAPs and UAVs, the cache-hit occurs, otherwise, it is known as a hit-miss and the requested content is served by the main server. To decrease the video traffic load on the main server, UAV-based femtocaching strategies increase the number of requests directly served by FAPs and/or UAVs [108]. However, due to the large size of video files, it is not feasible to store all contents in the storage of caching nodes. Additionally, taking into account the varying popularity of video contents over time and the constraint of cache space in both UAVs and FAPs, caching nodes' contents are periodically updated in the replacement phase according to the Fairness Scheduling algorithm with an Adaptive Time Window (FS-ATW) presented in [67].

In this work, by focusing on the delivery phase, the main goal is to propose a multi-objective UAV-based femtocaching strategy based on a real dense network to determine how to select caching nodes by ground users in order to enhance the QoS, in terms of user's access delay, minimize the UAVs' energy consumption and handover phenomena. In what follows, we briefly introduce some concepts required to develop the proposed CQN-CS framework. In this work, it is assumed that all ground users randomly move in all directions with a specific velocity, that will be introduced in "User Mobility Pattern". Then, we will fully describe four alternative ways to support users' requests in "User Access Pattern". A summary of the notations used hereinafter is provided in Table 3.1.

User Mobility Pattern

In this work, the Difference Correlated Random Walk (DCRW) is considered to model the ground users' movement pattern which is a common movement pattern in MEC networks [31, 109], where the current position of GU_j at time slot t, denoted by $\mathbf{L}_j(t) = [x_j(t), y_j(t)]^T$, depends on the movement's velocity. According to bivariate Stochastic Differential Equation (SDE), a ground user's velocity at time slot t, denoted by $\boldsymbol{v}_j(t) = [v_j^{(x)}(t), v_j^{(y)}(t)]^T$, is calculated as [109]

$$d\boldsymbol{v}_j(t) = -\begin{pmatrix} -\log \varsigma_1 & \theta \\ -\theta & -\log \varsigma_2 \end{pmatrix} (\boldsymbol{v}_j(t) - \boldsymbol{\mu})dt + \mathbf{J}d\mathbf{B}_t, \tag{2}$$

where ς_1 and ς_2 denote auto-correlation parameters in (x,y) coordinates, respectively, $\mathbf{J}$ denotes a (2×2) lower triangular matrix with positive diagonal components to determine the covariance of velocity shifts, and $\mathbf{B}_t$ represents a (2×1) vector to illustrate the standard Brownian motion at time slot t. In addition, $\boldsymbol{\mu}$ and θ signify the mean velocity vector and the DCRW model's mean turning angle, respectively. Accordingly, the location of the ground user GU_j at time slot t is obtained as [109]

$$\mathbf{L}_j(t) = \mathbf{L}_j(t-1) + \boldsymbol{\upsilon}_j(t-1)\Delta t, \tag{3}$$

where Δt represents the time interval between two consecutive estimated locations. To determine the initial location of ground users, we utilize the Angle of Arrival (AoA) scheme [10, 11], which has been recognized as an efficient and high accurate triangulation localization method among all the available schemes (e.g., see [15, 16]). By assuming the known position of FAPs, the initial location of ground user GU_j at time $t = 0$, denoted by $\mathbf{L}_j(0) = [x_j(0), y_j(0)]^T$, is obtained as follows

$$x_j(0) = \frac{d_{n,i} \tan\theta_{i,j}}{\tan\theta_{i,j} - \tan\theta_{n,j}}, \tag{4}$$

$$y_j(0) = \frac{d_{n,i} \tan\theta_{n,j} \tan\theta_{i,j}}{\tan\theta_{i,j} - \tan\theta_{n,j}}, \tag{5}$$

where $\theta_{i,j}$ and $\theta_{n,j}$ represent the angle between x-axis and the line between ground user GU_j and FAPs f_i and f_n, respectively [11]. Moreover, $d_{n,i}$ denotes the distance between i^{th} and n^{th} FAPs.

Users' Access Pattern

Given the location of users, all available FAPs and UAVs in the vicinity of ground users are determined. Therefore, by requesting content by ground user GU_j, this request must be served by one of the FAPs or UAVs in the neighborhood. In some cases, however, the requested content cannot be found in the storage of either available FAPs or UAVs, due to the limited embedded storage of caching nodes. Consequently, they need to provide the corresponding content from the cloud server. To serve users' requests, there are four alternative approaches, i.e., (*i*) FAP-ground user; (*ii*) UAV-ground user; (*iii*) Cloud-FAP-ground user, and; (*iv*) Cloud-UAV-ground user links. It should be noted that the last two approaches would happen when the requested content does not exist in the storage of either FAPs or UAVs.

This completes the description of the network's model and the main assumptions required for the development of the proposed CQN-CS framework. Next, we construct a multi-objective optimization problem over which the CQN-CS framework is designed.

3.1.2 Multi-Objective Optimization Problem

In this subsection, we present users' access delay, energy consumption of UAVs, and handover of FAPs in order to formulate a multi-objective optimization problem. The constructed optimization problem is associated with the selection of an appropriate UAV and/or FAP among all candidate caching nodes to serve users' requests.

Users' Access Delay

By considering the common consideration in the literature that all video files in the network are of similar size [108, 114], users' access delay exclusively depends on the connection type, i.e., the user's request is served by the FAP or the UAV. Note that this assumption is only used for notational convenience as larger contents in caching-based networks could be broken into the same length packets by packetization. Additionally, another key factor that has a great impact on the user's access delay is the availability of the requested content in the storage of FAPs and/or UAVs. If the requested content can be found in the cache of FAP and/or UAV, this request is directly handled by the caching node and the cache-hit occurs, otherwise, the corresponding file must be provided by the cloud server for FAP and/or UAV to manage the request, and in this case, the ground user will experience much more delay. As will be described shortly, the user's access delay is expressed as a function of the distance between the ground user and the target caching node, and the popularity of the requested file.

Users' Access Delay Through UAV Connection: To provide content via a UAV, an air-to-ground connection from the UAV to the ground user must be formed. In such a case that the requested content is not accessible in the UAV cache, in addition to the air-to-ground connection, a ground-to-air connection requires to be established as well, which is the wireless fronthaul link between the cloud server and the UAV. To determine the user's access delay through UAVs, due to the unavoidable obstacles throughout the network, we consider both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) connections over the Cloud-to-UAV and the UAV-to-Ground user links as follows:

- *UAV-to-Ground User Link:* The LoS and NLoS path loss models from UAV u_k to ground user GU_j at time slot t are expressed as follows [110]

  $$\mathcal{L}_{k,j}^{(LoS)}(t) = \mathcal{L}_0 + 10\eta^{(LoS)} \log(d_{k,j}(t)) + \chi_{\sigma}^{(LoS)}, \tag{6}$$

  $$\mathcal{L}_{k,j}^{(NLoS)}(t) = \mathcal{L}_0 + 10\eta^{(NLoS)} \log(d_{k,j}(t)) + \chi_{\sigma}^{(NLoS)}, \tag{7}$$

  where $\mathcal{L}_0 = 20 \log \left(\frac{4\pi f_c d_0}{c} \right)$ represents the path loss in reference distance d_0, f_c is the carrier frequency, and c denotes the light speed. In addition, $\eta^{(LoS)}$ and $\eta^{(NLoS)}$ indicate the path loss exponents of LoS and NLoS, respectively, and $d_{k,j}(t)$ is the Euclidean distance between UAV u_k and ground user GU_j at time slot t. Moreover, $\chi_{\sigma}^{(LoS)}$ and $\chi_{\sigma}^{(NLoS)}$, as the shadowing effects, denote zero-mean Gaussian-distributed random variables with the standard deviation σ. Considering the fact that the probability of LoS connection relies on the environment, the probability of LoS link, denoted by $p_{k,j}^{(LoS)}(t)$, is expressed as [111]

  $$p_{k,j}^{(LoS)}(t) = \left(1 + \psi \exp\left(-\zeta[\phi_{k,j}(t) - \psi]\right)\right)^{-1}, \tag{8}$$

  where ψ and ζ are environmental constant parameters, and $\phi_{k,j}(t) = \sin^{-1}\left(\frac{h_k}{d_{k,j}(t)}\right)$ represents the elevation angle, where h_k is the altitude of the UAV u_k. In our optimization problem, we assume that each UAV u_k flies in a fixed altitude h_k over the hovering time. As a result, the average path loss between UAV u_k and ground user GU_j is calculated as

$$\overline{\mathcal{L}}_{k,j}(t)=p_{k,j}^{(LoS)}(t)\mathcal{L}_{k,j}^{(LoS)}(t)+(1-p_{k,j}^{(LoS)}(t))\mathcal{L}_{k,j}^{(NLoS)}(t). \tag{9}$$

- *Cloud-to-UAV Link:* Similarly, since the terrain knowledge is not available, the link between UAV and the cloud cannot deterministically be identified as LoS or NLoS. Therefore, we use probabilistic mean path loss with two possible events (i.e., LoS or NLoS). The average path loss over the LoS and NLoS conditions is then computed by considering $p_{c,k}^{(LoS)}(t)$ denoting the probability of having a LoS, and $1-p_{c,k}^{(LoS)}(t)$ representing the probability of having a NLoS link as follows

  $$\overline{\mathcal{L}}_{c,k}(t)=p_{c,k}^{(LoS)}(t)\mathcal{L}_{c,k}^{(LoS)}(t)+(1-p_{c,k}^{(LoS)}(t))\mathcal{L}_{c,k}^{(NLoS)}(t), \tag{10}$$

  where $\mathcal{L}_{c,k}^{(LoS)}(t)=d_{c,k}^{-\alpha}(t)$ and $\mathcal{L}_{c,k}^{(NLoS)}(t)=\beta\mathcal{L}_{c,k}^{(LoS)}(t)$, in which $d_{c,k}(t)$ denotes the distance between the cloud server and UAV u_k, and α and β represent the path loss exponent and the additional path loss of the NLoS connection, respectively [115].

To express the user's access delay through UAVs, first we calculate the cache-hit probability, i.e., the probability of serving a request by UAV u_k at time slot t, denoted by $p_k^{(h)}(t)$, and cache-miss probability, denoted by $p_k^{(m)}(t)$, as follows

$$p_k^{(h)}(t)=\sum_{l\in\mathcal{C}_k}p_l(t)\leq 1, \tag{11}$$

$$p_k^{(m)}(t)=1-p_k^{(h)}(t), \tag{12}$$

where $\mathcal{C}_k$ denotes a set of contents stored in the cache of UAV u_k. We use the fact that the popularity of contents $p_l(t)$ in Eq. (11) is changing over time. Following a similar concept used to form the average path loss in Eq. (10), there are two possible events to define the user's access delay through UAV, i.e., the cache-hit event (which can happen with probability of $p_k^{(h)}(t)$) and the cache-miss event (which occurs with probability of $p_k^{(m)}(t)$). Accordingly, we have

$$\mathcal{D}_u(t)=p_k^{(h)}(t)\mathcal{D}_u^{(h)}(t)+p_k^{(m)}(t)\mathcal{D}_u^{(m)}(t), \tag{13}$$

where $\mathcal{D}_u^{(h)}(t)$, known as the cache-hit delay through UAV, is expressed as

$$\mathcal{D}_u^{(h)}(t) = \frac{L_c}{R_{k,j}} = L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, \mathbf{u}_{-k}) + N_0}\right), \tag{14}$$

with L_c, $R_{k,j}$, P_k, and N_0 denoting the size of file c_l, the data rate of the transmission from UAV u_k to GU_j, the constant transmit power of UAV u_k, and the noise power, respectively. Moreover, $I_k(t, \mathbf{u}_{-k})$ indicates the interference from other UAV-to-Ground user links except for the corresponding u_k link. Similarly, the cache-miss delay, denoted by $\mathcal{D}_u^{(m)}(t)$, is calculated as

$$\begin{aligned}\mathcal{D}_u^{(m)}(t) = L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{c,k}(t)/10}}{I_k(t, \mathbf{u}_{-k}) + N_0}\right) + \\ L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, \mathbf{u}_{-k}) + N_0}\right),\end{aligned} \tag{15}$$

where the first term illustrates the user's access delay associated with the Cloud-to-UAV link, while the second term on the Right Hand Side (RHS) of Eq. (15) is related to the UAV-to-Ground user link.

User's Access Delay Through FAP Connection: Another promising approach to provide the desired content for ground users is serving the request by FAPs, which leads to a ground-to-ground connection type between FAPs and users. Similarly, if the requested content cannot be found in the cache of neighboring FAPs, the corresponding content is transmitted by the cloud server to the FAP, and then it is sent to the ground user. Therefore, the user's access delay through the FAP connection is defined as follows

$$\mathcal{D}_f(t) = p_k^{(h)}(t)\mathcal{D}_f^{(h)}(t) + p_k^{(m)}(t)\mathcal{D}_f^{(m)}(t), \tag{16}$$

where $\mathcal{D}_f^{(h)}(t)$, as the cache-hit delay through FAPs, is expressed as

$$\mathcal{D}_f^{(h)}(t) = L_c \log^{-1}\left(1 + \frac{P_i|\tilde{\mathcal{H}}_{i,j}(t)|^2}{I_i(t, \mathbf{f}_{-i}) + N_0}\right), \tag{17}$$

where $\tilde{\mathcal{H}}_{i,j}(t) = \dfrac{h_{i,j}(t)}{\sqrt{\mathcal{L}_{i,j}(t)}}$ illustrates the fading channel effect with path loss between FAP f_i and ground user GU_j at time slot t, where $g_{i,j}(t) = |h_{i,j}(t)|^2$ represents the power gain of the short-term fading channel coefficient $h_{i,j}(t)$, which is a complex zero-mean Gaussian random variable with the standard deviation equals to one, and $\mathcal{L}_{i,j}(t)$ denotes the corresponding path loss. In addition, $I_k(t, \mathbf{f}_{-i})$ represents the interference from other FAP-user links excluding the corresponding f_i link.

Energy Consumption of UAVs

The energy consumption of UAV u_k, due to transmitting file c_l with the size of L_c to ground user GU_j, is calculated as [112]

$$E_{u_k}^{(LoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(LoS)}(t)(\tau_f - \tau_p), \tag{18}$$

$$E_{u_k}^{(NLoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(NLoS)}(t)(\tau_f - \tau_p), \tag{19}$$

where τ_f and τ_p represent the flyby and the pause times of UAV u_k, respectively. Moreover, $P_T(t)$, $P_R(t)$, and $P_j(t)$ denote the transmission and reception powers of 1 Mb file sharing, and the received power at ground user GU_j, respectively, where $P_j^{(LoS)}(t)$ and $P_j^{(NLoS)}(t)$ are calculated as

$$P_j^{(LoS)}(t) = P_0^{(LoS)} - 10\eta^{(LoS)} \log\left(\frac{d_{k,j}(t)}{d_0}\right) + \chi_\sigma^{(LoS)}, \tag{20}$$

$$P_j^{(NLoS)}(t) = P_0^{(NLoS)} - 10\eta^{(NLoS)} \log\left(\frac{d_{k,j}(t)}{d_0}\right) + \chi_\sigma^{(NLoS)}, \tag{21}$$

where $P_0^{(LoS)}$ and $P_0^{(NLoS)}$ represent the received power at distance d_0 in LoS and NLoS models, respectively. Consequently, the average energy consumption of UAV u_k is

$$\overline{E}_u(t) = p_{k,j}^{(LoS)}(t) E_{u_k}^{(LoS)}(t) + (1 - p_{k,j}^{(LoS)}(t)) E_{u_k}^{(NLoS)}(t). \tag{22}$$

Handover of FAPs

Dynamic UAV-based femtocaching networks, consisting of massively dense FAPs with small transmission ranges, are exposed to triggering frequent handovers during users' movements. By considering $RSSI_{i,j}(t)$ as the received signal strength by ground user GU_j from FAP f_i, we have

$$RSSI_{i,j}(t) = RSSI_0 + 10\eta \log_{10} \big(\frac{d_{i,j}(t)}{d_0}\big) + \chi_\sigma, \tag{23}$$

where $d_{i,j}(t)$ and d_0 denote the distance between FAP f_i and GU_j, and the reference distance equal to 1 m, respectively. Moreover, η represents the path loss exponent, which is 10 dB or 20 dB, and χ_σ is a zero-mean Gaussian with the standard deviation σ that represents the effect of shadowing in our femtocaching scheme [113]. During movement of the ground user GU_j, once $RSSI_{i,j}(t)$ drops below the threshold level P_{th}, defined as the minimum signal strength that can be detected by ground users, handover triggers, and GU_j connects to another neighboring FAP with the strongest signal. Since the received signal strength depends on the distance between the ground user GU_j and FAP f_i, the low value of $RSSI_{i,j}(t)$ indicates that GU_j is far from FAP f_i, leading to triggering handover within the shortest possible time. Moreover, taking into account a dynamic femtocaching network, where ground users move consistently in the coverage area of FAPs, it is essential to consider that the ground user GU_j is becoming close to the corresponding FAP or moving farther away during its movement. Hence, we define $\Delta_{i,j}(t)$ as follows

$$\Delta_{i,j}(t) = RSSI_{i,j}(t) - RSSI_{i,j}(t-1), \qquad i = 1, \ldots, \mathcal{U}_j^{(f)}, \tag{24}$$

where $\mathcal{U}_j^{(f)}$ denotes a set of accessible FAPs for the ground user GU_j. In this case, $\Delta_{i,j}(t) = 0$, $(1 \leq i \leq \mathcal{U}_j^{(f)})$, indicates that the ground user GU_j is a stationary user, therefore, it should be connected to the FAP with the highest value of $RSSI$. $\Delta_{i,j}(t) > 0$ shows that the ground user GU_j is becoming close to FAP f_i, while $\Delta_{i,j}(t) < 0$, which is the worst case scenario, shows that GU_j is moving far away from FAP f_i. In order to decrease the number of handovers, after requesting a content by the ground user GU_j, the RSSI value of all candidate FAPs in the vicinity of GU_j is measured and the target FAP f_i, with the highest value of handover $\mathcal{HO}_{i,j}(t)$ is selected to serve the user's request, where $\mathcal{HO}_{i,j}(t)$ is obtained as follows

$$\mathcal{HO}_{i,j}(t) = RSSI_{i,j}(t) + \Delta_{i,j}(t). \tag{25}$$

Problem Formulation

Based on the above derivations and considering the system model described in Subsection 3.1.1, the goal here is to develop an efficient scheduling connection to assign an appropriate caching node (i.e., UAV or FAP) to ground users. To construct the optimization problem, we consider the following three objectives: (i) User's access delay; (ii) Energy consumption of UAVs, and; (iii) Handover phenomena. On the other hand, to evaluate the efficiency of the proposed CQN-CS framework, we use the following six performance metrics: (1) Cache-hit ratio; (2) User's access delay; (3) Energy consumption of UAVs; (4) Handover; (5) Lifetime of the network, and; (6) Cumulative rewards. Toward this goal, we formulate a multi-objective optimization problem with the aim of minimizing the user's access delay depending on whether the user is served through FAPs or UAVs. Access delay experienced by the j^{th} ground user, energy consumption of UAVs, and the occurrence of handovers are denoted by $\mathcal{D}_{l,j}(t) \in \{\mathcal{D}_{f_i,j}(t), \mathcal{D}_{u,j}(t)\}$, $\overline{E}_{l,j}(t)$, and $\mathcal{HO}_{l,j}(t)$, respectively. Note that subscript $l \in \mathcal{U}_j = \{u_k, f_1, \ldots, f_{\mathcal{U}_j^{(f)}}\}$ denotes all caching nodes in the vicinity of the ground user GU_j. By considering the system model in Fig. 3.1, our proposed wireless network consists of some clusters, where each cluster is supported by one UAV and several overlapped FAPs. Consequently, the ground user GU_j (for $1 \leq j \leq N_g$) at time slot t can access one UAV u_k, (for $1 \leq k \leq N_u$), and several FAPs f_i, $i = 1, \ldots, \mathcal{U}_j^{(f)}$. Therefore, the cardinality of all caching nodes (UAV and FAPs) supporting the ground user GU_j at time slot t is given by $|\mathcal{U}_j| = |\mathcal{U}_j^{(f)}| + 1$. One of the most efficient methods to scalarize a set of objectives into a single objective, is the Weighted Sum (WS) method, in which normalized objectives are pre-multiplied by weights ω_q, and are combined as follows

$$\begin{aligned}\mathcal{F}_j(\mathbf{x}) &= \sum_{l\in\mathcal{U}_j}\big(\omega_1 x_l \mathcal{D}_{l,j}(t) + \omega_2 x_l \overline{E}_{l,j}(t) \\ &- \omega_3 x_l \mathcal{HO}_{l,j}(t)\big),\end{aligned} \tag{26}$$

where $\mathcal{F}_j(\mathbf{x})$ denotes the cost function associated with the ground user GU_j, and $\mathbf{x} = [x_1, \ldots, x_{\mathcal{U}_j}]$ is an indicator vector, where x_l would be 1 if l^{th} caching node serves the request of the ground user GU_j, otherwise it equals to 0. Note that x_1 illustrates the connection between GU_j and UAV u_k. Moreover, the weight coefficient ω_q should be valued in such a way that the higher value of ω_q indicates the superiority of that parameter, where $\sum\limits_{q=1}^{3}\omega_q = 1$. Since the dimension of these three parameters is not the same, we assume normalized values, where the delay associated with UAVs and FAPs are normalized by the maximum tolerable delay of UAVs ($\mathcal{D}_u^{(max)}$) and that of FAPs ($\mathcal{D}_f^{(max)}$), respectively. Similarly, the energy consumption of UAVs and the handover parameter are normalized by E_{max} and P_{th}, respectively. More precisely, by considering the fact that ground users will experience maximum latency in the boarder of transmission range of both UAVs and FAPs, $\mathcal{D}_u^{(max)}$ and $\mathcal{D}_f^{(max)}$ can be determined according to Eqs. (13) and (16), where $d_{k,j}(t) = R_u$ and $d_{i,j}(t) = R_f$, respectively. Similarly, E_{max} is obtained according to Eq. (22), where $d_{k,j}(t) = R_u$ in Eqs. (20) and (21).

Our aim is to minimize the objective function $\mathcal{F}_j(\mathbf{x})$. Note that the minus sign for the handover parameter in Eq. (26) is due to the fact that the RSSI value linearly depends on the $\mathcal{HO}_{l,j}(t)$ in Eq. (25). Therefore, connecting to the FAP with the highest value of $\mathcal{HO}_{l,j}(t)$ leads to decreasing the number of handovers. In this case, we expand $\mathcal{D}_{l,j}(t)$ in Eq. (26) as

$$\mathcal{D}_{l,j}(t) = \mathcal{D}_{u,j}(t) + \sum_{l\in\mathcal{U}_j^{(f)}}\mathcal{D}_{f_l,j}(t). \tag{27}$$

Then, we can expand the objective function, including two terms; UAV connection, and FAP connection, as follows

$$\begin{aligned}\mathcal{F}_j(\mathbf{x}) = &\underbrace{\omega_1 x_1 \mathcal{D}_{u,j}(t) + \omega_2 x_1 \overline{E}_{u,j}(t)}_{\triangleq \mathbf{C_{U-G}}} + \\ &\underbrace{\sum_{l \in \mathcal{U}_j^{(f)}} \omega_1 x_l \mathcal{D}_{f_l,j}(t) - \omega_3 x_l \mathcal{HO}_{l,j}(t),}_{\triangleq \mathbf{C_{F-G}}}\end{aligned} \tag{28}$$

where $\mathbf{C_{U-G}}$ denotes the cost function associated with UAVs and the ground users' connections, and $\mathbf{C_{F-G}}$ is the connection link between FAPs and the ground users. Consequently, $\mathbf{C_{U-G}}$ would be zero, if the ground user GU_j should be served by FAPs. Similarly, $\mathbf{C_{F-G}}$ would be zero, when GU_j is served by UAV u_k. To determine the time varying connection scheduling of ground users, by considering the fact that there are N_g ground users in the network, the proposed scalarized optimization problem is expressed as follows [118]

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{j \in N_g} \mathcal{F}_j(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{C1.}\ \ x_l \in \{0,1\}, \\ & \mathbf{C2.}\ \ \sum_{l=1}^{\mathcal{U}_j} x_l = 1, \\ & \mathbf{C3.}\ \ 0 \leq \mathcal{D}_{u,j}(t) \leq 1, \\ & \mathbf{C4.}\ \ 0 \leq \mathcal{D}_{f_l,j}(t) \leq 1, \\ & \mathbf{C5.}\ \ 0 \leq \overline{E}_{u,j}(t) \leq 1, \\ & \mathbf{C6.}\ \ RSSI_{i,j}(t) \geq P_{th}. \end{aligned} \tag{29}$$

For the above optimization problem, x_l in constraint **C1** is the indicator variable. Constraint **C2** represents that the request of each ground user is served by one caching node. Constraints **C3** $\sim$ **C5** are utilized to illustrate all possible caching nodes in the vicinity of the ground user GU_j, where the normalized latency associated with candidate UAVs and FAPs, and the energy consumption constraints should be positive and less than or equal to one. Moreover, constraint **C6** represents that the RSSI value should be equal to or greater than the threshold level of RSSI (P_{th}), otherwise handover occurs. Following a similar argument as in [114], our minimization problem in Eq. (29) can be expressed as a maximization problem, i.e.,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{j \in N_g} \mathcal{F}'_j(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{C1} \sim \mathbf{C6}, \end{aligned} \tag{30}$$

where $\mathcal{F}'_j(\mathbf{x})$ is defined as

$$\begin{aligned} \mathcal{F}'_j(\mathbf{x}) = \omega_1 x_1 \Big(1 - \mathcal{D}_{u,j}(t)\Big) + \omega_2 x_1 \Big(1 - \overline{E}_{u,j}(t)\Big) \\ + \sum_{l \in \mathcal{U}_j^{(f)}} \omega_1 x_l \Big(1 - \mathcal{D}_{f_l,j}(t)\Big) + \omega_3 x_l \mathcal{HO}_{l,j}(t). \end{aligned} \tag{31}$$

This completes our formulation of a multi-objective optimization problem to present the user's access delay, energy consumption of UAVs, and the handover of FAPs. Next, we develop the proposed CQN-CS scheduling architecture based on the developed multi-objective optimization formulation.

3.1.3 The CQN-CS Scheduling Framework

In this Subsection, we present an optimum framework, the CQN-CS, to identify how users access UAVs and/or FAPs based on the RL model, in order to solve the optimization problem of Eq. (30). To be specific, we first briefly introduce the required background on RL, then we present the proposed CQN-CS, which is an efficient DQN model with an embedded CNN connection scheduling architecture developed for a UAV-based femtocaching network.

RL Background

Generally speaking, RL algorithms consist of an agent, interacting with an environment based on a set of given actions. The agent receives feedback, as a reward or punishment, from the environment after each interaction, and updates its states accordingly. Markov Decision Process (MDP) provides the rigorous mathematical foundation for RL algorithms, and includes a set of $\mathcal{A}$ actions, a set of states $\mathcal{S}$, a transition function $\mathcal{T}$, and a reward function, denoted by $\mathcal{R}$. Each action $\boldsymbol{a}_t \in \mathcal{A}$ at time slot t in any state $\boldsymbol{s}_t \in \mathcal{S}$ results in a new state $\boldsymbol{s}_{t+1} \in \mathcal{S}$ at time slot $t+1$ based on the transition function $\mathcal{T}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ and a reward $r_t = \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The aim of MDP is to find the optimum policy π^* to achieve the maximum accumulated rewards obtained over an infinite number of interactions [116], where π^* is expressed as follows

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\Big\{\sum_{t=0}^{H-1} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}\Big\}, \tag{32}$$

where H denotes the number of finite episodes in MDP and $\gamma \in [0, 1]$ is the discount factor. The low value of γ leads to maximizing short-term rewards, while a high value of γ increases rewards over a longer period of time.

The Q-Network framework, as one of the most commonly used value-based and model-free RL algorithms, can be considered as a function approximator, where the value of $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ relies on action $\boldsymbol{a}_t$ and the state $\boldsymbol{s}_t$ of the agent at time slot t, expressed as follows [116]

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}_{\pi}\Big\{\sum_{t=0}^{H-1} \gamma^t r_{t+1} | \boldsymbol{s}_0 = \boldsymbol{s}, \boldsymbol{a}_0 = \boldsymbol{a}, \boldsymbol{a}_t = \pi(\boldsymbol{s}_t)\Big\}. \tag{33}$$

In this regard, the value of $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ in each time slot is updated by the agent as follows

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) \leftarrow (1-\lambda) Q(\boldsymbol{s}_t, \boldsymbol{a}_t) + \lambda(r_t + \gamma \max Q(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})), \tag{34}$$

where $\lambda \in [0, 1]$ is the learning rate. If the number of states is finite, the Q-learning approach performs efficiently to update the state-action value function in each state. In scenarios, where the number of states is infinite, however, it is not feasible to visit all the states, therefore, deep learning methods can contribute to approximate the state-action value function. In deep Q-learning approaches, a deep model is used for prediction and training, instead of building a Q-table to look up and update values. In this work, we apply CNN, as one of the most efficient deep learning methods, to estimate Q-values.

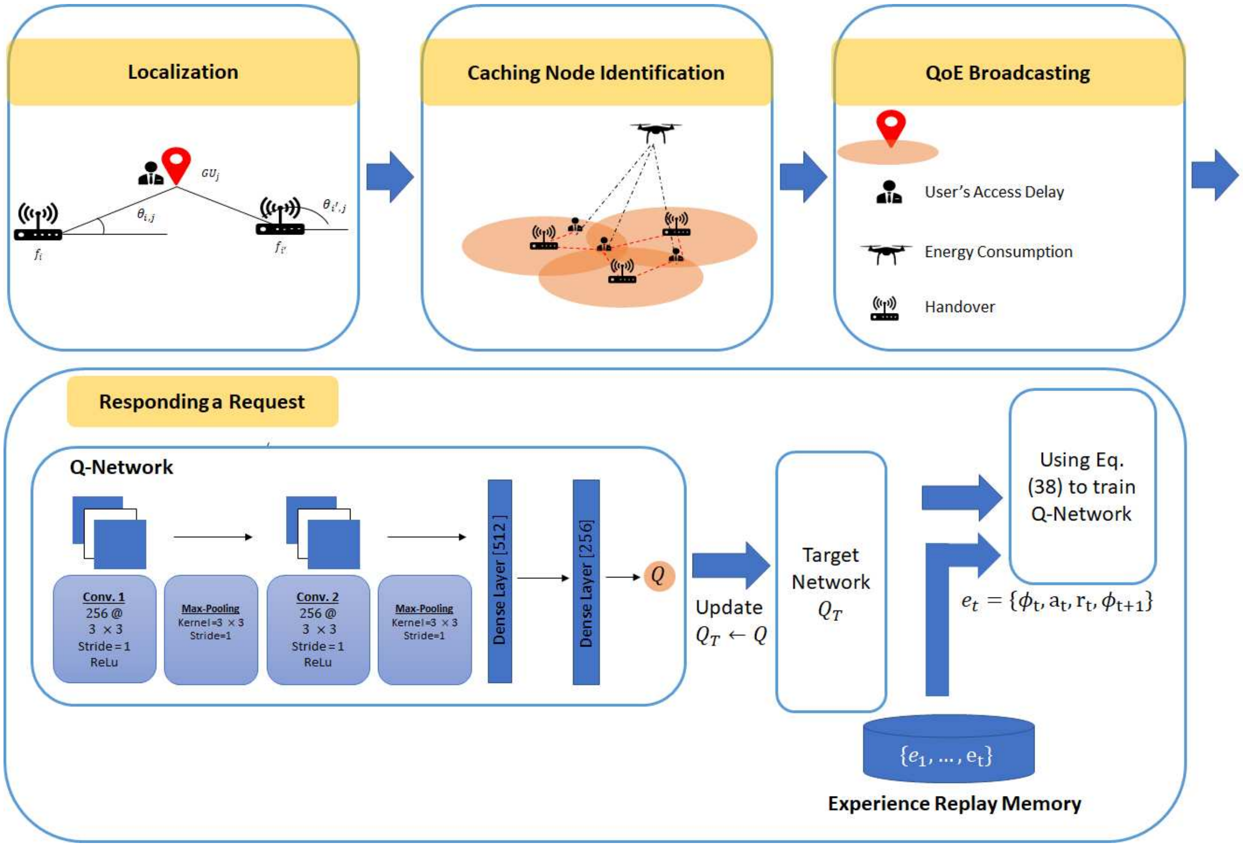


Figure 3.2: The block diagram of our proposed CQN-CS framework.

The CQN-CS Architecture

Due to the dynamic behavior of the UAV-based femtocaching network, which is a result of the mobility of ground users in the environment, we train our CQN-CS model using QoS requirements, including users' access delay, and the QoE from UAV and FAP perspectives. Fig. 3.2 illustrates the block diagram of our proposed CQN-CS framework. By considering a slotted structure, each ground user learns how to access the UAV and/or FAP autonomously, where each time slot t consists of the following four steps:

Step 1 (Localization): To determine the best caching node for serving a request, we need to know all possible caching nodes in the vicinity of ground users. Toward this goal and in the first stage, the location of ground users must be estimated. According to the AoA localization technique, the location of each user is calculated based on Eqs. (4) and (5).

Step 2 (Caching Node Identification): In this phase and according to the constraint **C2** in Eq. (29), we need to determine all FAPs and UAVs in the vicinity of the ground user GU_j to build $\mathcal{U}_j = \{u_k, f_1, \ldots, f_{\mathcal{U}_j^{(f)}}\}$. Accordingly, the distance between FAP f_i and the ground user GU_j is calculated as follows

$$\sqrt{(x_j - x_{f_i})^2 + (y_j - y_{f_i})^2} \leq R_f, \tag{35}$$

where (x_j, y_j) and (x_{f_i}, y_{f_i}) represent spatial coordinates of GU_j and FAP f_i, respectively. Therefore, all ground users that are positioned in the transmission range of FAP f_j can be supported by the corresponding FAP. Similarly, to calculate the distance between UAV u_k and the ground user GU_j, we have

$$\sqrt{(x_j - x_{u_k})^2 + (y_j - y_{u_k})^2 + h_{u_k}^2} \leq R_u, \tag{36}$$

where $(x_{u_k}, y_{u_k}, h_{u_k})$ is the spatial coordinate of UAV u_k. Since the height of ground users is much lower than the altitude of flying UAVs, the height of ground users is negligible. Note that the above statements are equivalent to constraints **C3** $\sim$ **C5**. According to Eqs. (35) and (36), the set of available candidate caching nodes for the ground user GU_j, denoted by $\mathcal{U}_j$, is built.

Step 3 (QoE Broadcasting): Given $\mathcal{U}_j$, constructed from Step 2 above, all FAPs and UAVs in the vicinity of the ground user GU_j broadcast the energy consumption and the probability of handover, calculated based on Eqs. (22) and (25). Considering these decision criteria, consequently, results in the minimization of the user's access delay in the next time slots, obtained according to Eqs. (13) and (16). Then, the cost of selecting FAPs and UAVs is calculated by ground users based on the QoE, and air-to-air and air-to-ground channel path loss models. Consequently, all parameters in Eq. (30) are known. In the next stage, ground users will select the target caching node, either UAVs or FAPs, in order to maximize Eq. (30). Accordingly, to satisfy constraints **C1** and **C2**, only x_l associated with the target caching node would be 1.

Step 4 (Responding a Request): To solve Eq. (30), we propose a CNN-based Q-learning approach in the context of the UAV-based femtocaching network. Our CQN-CS framework, represented as a tuple $\{\boldsymbol{s}_t, \boldsymbol{a}_t, r_t\}$, has the following components:

(i) CQN-CS Agents: In our proposed CQN-CS framework, ground users, as the intelligent members of our problem, act as agents.

(ii) CQN-CS Action Space: The CQN-CS architecture has an action generation engine to find a globally optimized policy based on the previous states to learn the most suitable action. Here, the action refers to the selection of a suitable caching node to support ground users' requests in various circumstances in terms of the location of ground users, the battery life of UAVs, channels' condition, and the probability of handover. After requesting a content by user GU_j, this request should be served by one of the caching nodes in the vicinity of GU_j, denoted by $l \in \mathcal{U}_j = \{u_k^{(j)}, f_1^{(j)}, \dots, f_{\mathcal{U}_j^{(f)}}^{(j)}\}$, where superscript (j) , $j = 1, \dots N_g$, indicates all ground users in the network. Therefore, all UAVs and FAPs in this set can be an action for GU_j, i.e., $\boldsymbol{a}_t = [u_k^{(1)}, f_1^{(1)}, \dots, f_{\mathcal{U}_1^{(f)}}^{(1)}, \dots, u_k^{(N_g)}, f_1^{(N_g)}, \dots, f_{\mathcal{U}_{N_g}^{(f)}}^{(N_g)}]^T$, for $(1 \leq k \leq N_u)$, which is equivalent to $\mathbf{x} = [x_1^{(1)}, \dots, x_{\mathcal{U}_1}^{(1)}, \dots, x_1^{(N_g)}, \dots, x_{\mathcal{U}_{N_g}}^{(N_g)}]^T$ in Eq. (30). Therefore, by selecting an action, the corresponding value of $x_l^{(j)}$ would be 1, otherwise it would be 0. Note that selecting the optimal action leads to maximizing the reward of the network.

(iii) CQN-CS State: The action is selected based on the current system state $\boldsymbol{s}_t$ at time slot t. In each time slot t, the value of user's access delay via UAVs and FAPs, energy consumption of UAVs, and handover represent states in our proposed framework, i.e., $\boldsymbol{s}_t = [\mathbf{L}_j(t), \mathcal{D}'_{u,j}(t), \overline{E}'_{u,j}(t), \mathcal{D}'_{f_l,j}(t), \mathcal{HO}'_{l,j}(t)]^T$. More specifically, $\boldsymbol{s}_t$ consists of the following five components:

- $\mathbf{L}_j(t)$: The location of ground user GU_j at time slot t, which is determined according to Eq. (3).
- $\mathcal{D}'_{u,j}(t)$: The total delay that the ground user GU_j will experience if its request is served by UAV u_k until time slot t.
- $\overline{E}'_{u,j}(t)$: The total energy consumed by UAV u_k at time slot t for establishing a connection with the ground user GU_j.
- $\mathcal{D}'_{f_l,j}(t)$: The total user's access delay GU_j due to the connection with FAP

f_l.

- $\mathcal{HO}'_{l,j}(t)$: The handover indicator of FAP f_l at the user side GU_j.

Therefore, each ground user should build a Q-table, where rows denote all possible states, and columns indicate actions. However, since the number of states is infinite, it is not possible to visit all the states. For this reason, we apply CNN on the Q-Network to estimate Q-values. It should be noted that by serving a request by FAPs, the energy consumption of UAVs has no change, therefore, the state value of the energy consumption would be constant. Moreover, since we assume that the handover phenomenon just occurs in FAPs and ground users' connection, there is no change in the state value of handover when a request is served by UAVs. In such a case that more than one ground user simultaneously selects a specific caching node, the corresponding caching node will randomly select one of them to support the request.

(iv) CQN-CS Reward: According to the optimization problem in Eq. (30), any reduction in the involved metrics (i.e., user's access delay via UAVs and FAPs, energy consumption of UAVs, and handover) results in a higher reward. When it comes to the problem of finding optimal action policies within the Pareto optimal set in scenarios with multiple and conflicting objectives, different multi-objective RL-based models [118,119,206] were considered in the literature. In this work, we employ the weighted-sum approach, where the underlying set of objectives is converted into a single function by assigning a pre-defined weight to each individual objective. More specifically, the reward function in our multi-objective RL-based framework, denoted by $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, is extended to a vector reward function, denoted by $\mathbf{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) = [\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t), \mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t), \mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t)]$, where $\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t)$, $\mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and $\mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t)$ represent the reward function associated with the users' access delay, energy consumption of UAVs, and FAPs' handover, respectively, calculated as follows:

$$\mathcal{R}_1(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} 1 - \mathcal{D}_{u,j}(t), & \text{UAV Link}, \\ 1 - \mathcal{D}_{f_l,j}(t), & \text{FAP Link}. \end{cases}, \tag{37}$$

$$\mathcal{R}_2(\boldsymbol{s}_t, \boldsymbol{a}_t) = 1 - \overline{E}_{u,j}(t), \tag{38}$$

$$\text{and} \quad \mathcal{R}_3(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathcal{HO}_{l,j}(t). \tag{39}$$

In this case, the weighted-sum form of the Q-value is calculated as

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \sum_{q=1}^{N_o=3} \omega_q Q_q(\boldsymbol{s}_t, \boldsymbol{a}_t), \tag{40}$$

where N_o denotes the number of incorporated objectives. Finally, the action associated with the largest $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is selected by an agent [119]. Considering the Pareto optimality, the trade-off is parameterized by the weight coefficients $\omega_q \in [0, 1]$, where $\sum_{q=1}^{N_o} \omega_q = 1$ [119]. In this case, higher values of ω_q indicate the superiority of that objective. In this work, we have three conflicting objectives, i.e., users' access delay, energy consumption of UAVs, and handover phenomena occurred between FAPs. To maintain a balance between the QoS of ground users and the QoE of FAPs and UAVs, the proposed connection scheduling framework aims to simultaneously satisfy concerns of ground users, UAVs, and FAPs. For this reason, we assign equal weights to the three underlying objectives. In this case, the policy is said to be Pareto optimal if the value of π^*, obtained according to Eq. (32), strictly dominates or is incomparable with the value functions of other policies [119].

After selecting a caching node with the largest $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the connection information associated with the corresponding action, the location of the ground user, the probability of handover, the energy consumption of UAVs, and the user's access delay are stored in the memory replay of the proposed CQN model. In our proposed framework, for each state-action pair, CQN approximates the Q-function by using CNN with tunable weight parameters, which is a non-linear approximator. The CNN model, however, needs to be retrained due to the mobility of ground users and the dynamic nature of UAVs (i.e., the battery life). Therefore, a replay memory is used for past experienced state-action pairs and the associated rewards. The weight of filters in each layer at time slot t is denoted by ξ_t. The observed state sequence, including β state-action pairs at time slot t, is denoted by $\boldsymbol{\phi}_t = [\boldsymbol{s}_{t-\beta}, \boldsymbol{a}_{t-\beta}, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{s}_t]$, which is the input of the CNN to estimate $Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_t)$. The experience memory pool is denoted by $D = \{e_1, \ldots, e_t\}$, where $\mathbf{e}_t = (\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1})$. The state sequence in replay buffer $\mathbf{e}_m$ is selected randomly to update the weight parameter ξ_t according to the Stochastic Gradient Descent (SGD) method. By choosing ξ_t, our goal is to minimize the loss function, denoted by $\mathbb{L}(\xi_t)$, which is the mean-squared error of the target optimal Q-function with the minibatch updates, given by

$$\mathbb{L}(\xi_t) = \mathbb{E}_{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}} \left[\left(Q_T - Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t | \xi_{t+1}) \right)^2 \right], \tag{41}$$

where Q_T is the target optimal Q-function, given by

$$Q_T = r_t + \gamma \max_{\boldsymbol{a}_t^{'}} Q(\boldsymbol{\phi}_{t+1}, \boldsymbol{a}_t^{'} | \xi_{t-1})). \tag{42}$$

Eventually, the action $\boldsymbol{a}_t$ is chosen for the state $\boldsymbol{s}_t$ based on the ϵ-greedy algorithm. With the probability of $(1 - \epsilon)$, the best action $\boldsymbol{a}_t^*$ is chosen from the set of Q-functions as follows

$$\boldsymbol{a}_t^* = \arg\max_{\boldsymbol{a}_t^{'}} Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t^{'}). \tag{43}$$

Then, the user's request should be served by the corresponding caching node with action $\boldsymbol{a}_t^*$. Accordingly, the reward r_t is calculated by agent, and the new experience $\{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}\}$ is stored in the replay memory by agent. The pseudo-code of our proposed CQN-CS framework is outlined in **Algorithm 1**. The rationale behind the design of **Algorithm 1** is described in more details based on the following steps:

- *Initialization:* In each epoch, all parameters are selected according to the values shown in Table 3.2. Moreover, we reset all the parameters related to the environment such as the replay buffer. These actions are equivalent to Lines 1-5 of Algorithm 1.

There are N_g number of ground users who are agents in the proposed network model. The following steps will be performed for all ground users in each episode, denoted by t.

- *Localization:* To construct the action space of each agent, the current location of ground users and the possible caching nodes in their vicinity are gathered. After initializing the location of ground users in the environment, the current location of ground users in each episode is determined according to Lines 10 and 11 of Algorithm 1.

- *Caching Node Identification:* After identifying the location of agents, all available caching nodes in the vicinity of ground users should be determined. In this case, the action space is constructed for each agent, which is equivalent to Lines 12 and 13 of Algorithm 1.
- *QoE broadcasting:* To update the state space of ground users, all UAVs and FAPs in the vicinity of each ground user broadcast their energy consumption and experienced handover, which is equivalent to Lines 14 and 15 of Algorithm 1.
- *Responding to a Request:* Considering the current state space, each agent selects the best caching node in such a way that its Q-value is maximized. If the value of the current episode is less than a pre-specified threshold β, a random action with a probability of ϵ is selected (Lines 18 and 19); otherwise, an optimal action is selected based on the ϵ-greedy policy (Lines 21-24). After taking an action, the state-space, the action-space, and the reward vector are updated and stored in the memory of the CQN model (Lines 25-32). For each state-action pair, CQN approximates the Q-function using CNN with tunable weight parameters (Lines 33-37).
- *Termination:* At the end, weight parameters are selected in such a way that the loss function, expressed in Eq. 41, is minimized. Consequently, the optimal action, which leads to increasing the Q-value, is selected by each agent at episode t.

Computational Complexity

As previously mentioned, due to the infinite number of states in the UAV-based femtocaching framework, the size of the action-state space observed by each ground user is relatively high. In such scenarios with a high number of state-action pairs, the computational cost of conventional Q-learning algorithms is significantly high. Therefore, Deep Q-Learning (DQN) models are typically used where instead of storing expected rewards associated with each state-action pair in a Q-table, a Deep Neural Network (DNN) model is used to select the actions according to the agent's current state [120]. In complex problems such as the one at hand, several information sources (such as the position of agents,

Algorithm 1 The CQN-CS Framework

```
Initialization:
Set all parameters according to Table 5.10.
Initialize state-action pairs $\beta$.
Initialize batch size $B$.
Reset replay memory size.
for $j = 1, \dots, N_g$ do
    for $t = 1, 2, \dots$ do
        Input:
        $s_0 = [\mathbf{L}_j(0), \mathcal{D}'_{u,j}(0), \overline{E}'_{u,j}(0), \mathcal{D}'_{f_l,j}(0), \mathcal{HO}'_{l,j}(0)]$
        Localization: Determine the location of ground
          user $GU_j$ based on Eqs. (4) and (5).
        Identification: Determine all possible caching
          nodes in the vicinity of $GU_j$.
        QoE broadcasting: Calculate and broadcast the
          energy consumption, handover, and delay.
        Responding to a Request: Select the best
          caching node according to the following steps
        if $t \leq \beta$ then
            Choose an action randomly $\boldsymbol{a}_t \in \{1, \dots, \mathcal{U}_j\}$
        else
            Obtain CNN output $Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t | \xi_t)$ with input
                $\boldsymbol{\phi}_t$ and weights $\xi_t$.
            Choose $\boldsymbol{a}_t$ via $\epsilon$-greedy algorithm
        end if
        Observe $\mathbf{L}_j(t), \mathcal{D}'_{u,j}(t), \overline{E}'_{u,j}(t), \mathcal{D}'_{f_l,j}(t), \mathcal{HO}'_{l,j}(t))$
        Estimate the Reward $r_t$ and obtain
            $s_{t+1} = [\mathbf{L}_j(t+1), \mathcal{D}'_{u,j}(t+1), \overline{E}'_{u,j}(t+1),$
          $\mathcal{D}'_{f_l,j}(t+1), \mathcal{HO}'_{l,j}(t+1))]$
        Create state, action, and reward vector:
          $\boldsymbol{\phi}_{t+1} = [\boldsymbol{s}_{t-\beta+1}, \boldsymbol{a}_{t-\beta+1}, \dots, \boldsymbol{a}_{t+1}, \boldsymbol{s}_{t+1}]^T$
        Add the new experience $\{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1}\}$
          to memory $D$
        for $d = 1, \dots, B$ do
            Select randomly $(\phi_d, \boldsymbol{a}_d, r_d, \phi_{d+1})$ from $D$
            Train CNN for $N$ iterations
            Calculate $Q_T$ using Eq. (42).
        end for
        Update the weight parameter $\xi_t$ using Eq. (41).
    end for
end for
Output: Optimal UAV/FAP connection scheduling with maximum $r_t$.
```

their movement directions, and available caching nodes) are simultaneously required to perform the action selection task. CNN architecture is an attractive solution to extract the relevant features from this pool of information. CNN-based architecture uses convolutional kernels to compress the state-space and extract temporal correlations between the current state of a ground user and previous state-action pairs. Within the CNN architecture, weights are shared between the episodes, which leads to a considerable reduction in the computational complexity. To compute the computational complexity of the proposed learning method, we follow the approach introduced in Reference [121] as CNN constitutes the main component (computational wise) of the proposed CQN-CS framework. The computational complexity of CNN depends on the number of multiplications in each convolutional layer [121]. Generally speaking, CNN models consist of $\mathcal{N}_l$ number of convolutional layers, where each layer includes F_l filters with size $W_l^f \times L_l^f$, zero-padded by P_l number of padding layers, and with stride size of S_l. Moreover, there are $\mathcal{N}_{fc}$ number of fully connected layers, including $\mathcal{N}_r$ number of Rectified Linear Units (ReLUs) to estimate the Q-value associated with each possible action. By considering the fact that the pooling and fully connected layers take only $5-10\%$ of the computational time [121], the computational complexity of CNN can be expressed as follows

$$\mathcal{N} = \sum_{l=1}^{\mathcal{N}_l} F_{l-1} W_l^f L_l^f F_l W_l^o L_l^o, \tag{44}$$

where l is the index of the convolutional layer. Terms F_{l-1} and F_l denote the number of input channels and the number of filters of l^{th} layer, respectively. In addition, W_l^o and L_l^o represent the width and the length of the output, calculated as follows

$$W_l^o = \frac{W_{l-1}^o - W_l^f + 2P_l}{S_l} + 1, \tag{45}$$

$$\text{and } L_l^o = \frac{L_{l-1}^o - L_l^f + 2P_l}{S_l} + 1, \tag{46}$$

where S_l and P_l are the size of stride and padding layers corresponding to the l^{th} layer, respectively. To calculate the computational complexity of the learning process of the proposed CQN-CS according to Eqs. (44)-(46), the value of $F_0 \times W_0^o \times L_0^o$ is equal to $\beta l_s \mathcal{U}$. In this case, β is the temporal memory depth, l_s is the length of the state space that is equal to 5, and $\mathcal{U}$ represents the number of possible actions in each episode.

3.1.4 Simulation Results

In this Subsection, we evaluate the performance of our proposed CQN-CS UAV-based femtocaching framework in terms of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, cumulative rewards, and the lifetime of the network. For the scenarios under simulation, we investigate how the CQN-based connection scheduling scheme affects the aforementioned performance metrics.

Simulation Setup: We consider an ultra-dense network with the radius $R =$ 5000 m, covered by a cloud server. In our proposed network, there are $N_f = 240$ FAPs with $R_f = 30$ m, uniformly distributed in the network, where each FAP overlaps with neighboring FAPs. Additionally, by considering the restrictions of the aviation regulations, we consider $N_u = 10$ UAVs flying horizontally at a specific altitude $h_k = 100$ m, for $(k = 1, \ldots, N_u)$ to cover a region of the network with $R_u = 500$ m. The area of interest is divided into K clusters according to the K-means algorithm, where $N_c \in \{20, 25, 30\}$ neighboring FAPs are covered by a UAV. Table 3.2 illustrates a list of parameters.

To consider the dynamic nature of the environment, we assume the Difference Correlated Random Walk (DCRW) model for the mobility pattern of ground users. By requesting a content by ground users, each UAV hovers at its location or flies a distance to manage the request, which both of them are energy consuming. However, since these concepts have been well studied in the literature of positioning management of UAVs, we only focus on the energy consumed by stationary UAVs to handle a request, depending on the distance between the UAV and the ground user, channel condition, and availability of the requested content in the cache of UAV. The proposed CQN-CS framework is performed for 100 epochs. By considering the fact that the battery life of UAVs is limited, we assume that each epoch is terminated if the total energy consumption of at least one UAV exceeds the UAV's battery life. Our CNN model consists of

Table 3.2: List of Parameters.

Notation	**Value**	**Notation**	**Value**
N_g	500	$\eta^{(LoS)}$, $\eta^{(NLoS)}$	2.5, 3 [21]
N_f	240	h_k	100 m [161]
N_u	10	α, β	2, 20 [108]
N_c	$\{20, 25, 30\}$	L_c	37.5 MB [67]
R_f	30 m [124]	τ_p	$0-5$ s [21]
R_u	500 m	P_k	15 dBm [108]
S_f	22.5 GB	$\chi_\sigma^{(LoS)}$, $\chi_\sigma^{(NLoS)}$	3.5, 3 [21]
S_u	22.5 GB	P_{th}	-67 dBm [25]
$\mathcal{C}$	184670	N_0	-94 dBm [108]
$P_T(t)$, $P_R(t)$	0.5 , 0.25 W [21]	γ	0.6 [123]

two 2-dimensional convolutional layers and two Fully Connected (FC) layers. Convolutional layers consist of 256 filters, each with the size of 3 and stride 1. We use max-pooling and ReLU as the activation function in each layer. The first FC layer consists of 512 ReLU units and the second FC has 256 ReLU units. To the best of our knowledge, there is no coupled UAV and femtocaching framework, studied from the connection scheduling perspective, for comparison purpose. Therefore, we introduce two baseline models for comparison:

- ***Q-Network Connection Scheduling (QN-CS) UAV-based Femtocaching Scheme:*** In this algorithm, the best caching node to handle users' requests is selected as the result of a Q-learning framework. All parameters of the RL approach, including actions, states, and rewards are kept similar to our proposed scheme for a fair comparison.
- ***Deep Q-Network Connection Scheduling (DQN-CS) UAV-based Femtocaching Scheme:*** Similar to the previous baseline, all the parameters are the same as our proposed framework, with the difference that in this baseline, we use Multilayer Perceptron (MLP) instead of CNN with two hidden layers, where each layer consists of 256 neurons.

Taking the above considerations into account, we illustrate the superiority of our proposed CQN-CS framework compared with conventional schemes from the aspect of the cache-hit ratio, user's access delay, energy consumption of UAVs, handover, cumulative rewards, and the lifetime of the network.

Performance Evaluation

We first evaluate the effectiveness of the AoA scheme as an efficient localization method to estimate the proximity of ground users, in order to determine possible caching nodes in their vicinity. Fig. 3.3 illustrates a typical 20×20 m^2 area, where ground users are randomly distributed. By Assuming (x_j, y_j) and $(\hat{x_j}, \hat{y_j})$ as the real and the estimated coordinates of user GU_j, respectively, the Root Mean Square Error (RMSE) is defined as $R_{RMSE} = \sqrt{\frac{1}{N_g}\sum_{j=1}^{N_g}(\hat{x_j} - x_j)^2 + (\hat{y_j} - y_j)^2}$ to evaluate the accuracy of location estimation. Taking into account the multipath and path loss effects, the RMSE of the AoA method in our proposed network is about 0.4 m, which is acceptable in comparison to the transmission range of FAPs.

We evaluate the convergence of the proposed CQN-CS framework in Fig. 3.4. More specifically, the convergence of the proposed CQN-CS framework is the crucial property to obtain a policy, which maps states to the optimal actions. According to Eq. 41, the main goal of the learning process is to minimize the loss function, which is the mean-squared error of the target optimal Q-function with the minibatch updates. Fig. 3.4 illustrates the convergence behavior of the proposed CQN-CS scheme. According to the result shown in Fig. 3.4, the CQN-CS framework converges after 80 epochs, which is an acceptable speed.

Cache-hit ratio is a metric used to express the number of requests served by caching nodes, either FAPs or UAVs, in each episode. We assume a pre-specified threshold for the battery life of UAVs, where reaching the energy consumption of at least one UAV to the threshold level is known as the *game over* in our RL network. The normalized cache-hit-ratio, denoted by $\mathcal{CH}^{(n)}$, consists of two terms, i.e., the satisfied requests served by FAPs, denoted by $\mathcal{CH}_f$, and the satisfied requests managed by UAVs, denoted by $\mathcal{CH}_u$, given by

$$\mathcal{CH}^{(n)} = \frac{\mathcal{CH}_f + \mathcal{CH}_u}{\mathcal{CH}_{max}}, \tag{47}$$

where $\mathcal{CH}_{max}$ is the maximum value of $\mathcal{CH}$ in all episodes. Fig. 3.5 evaluates the performance of our proposed scheme and other two baselines mentioned above from the aspect of the cache-hit ratio in different epochs. According

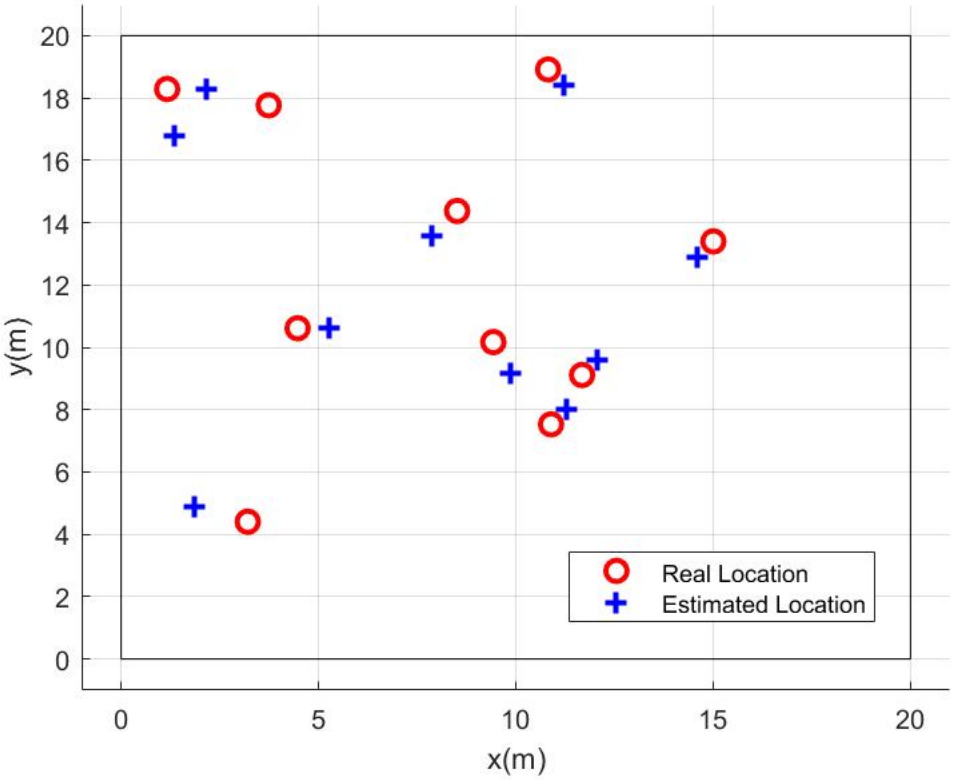


Figure 3.3: A typical location estimation result based on the AoA localization scheme.

to the results in Fig. 3.5, we can see that the area below the curve increases as the number of epochs grows. This is due to the fact that by passing the time, our network learns how to manage requests to expand the lifetime of UAVs. Moreover, our proposed CQN-CS method experiences more cache-hit-ratio, which indicates its superiority.

Fig. 3.6 illustrates the cumulative rewards of all caching nodes in each epoch before the energy consumption of at least one UAV reaches the battery life of the UAVs. According to the definition of rewards in our CQN-CS framework, connecting to the nearest FAP instead of the far one leads to a remarkable reduction in the user's access delay and handover, followed by increasing the reward value. UAV's connection would be efficient in such cases where there is no available FAPs, and/or when serving by UAVs leads to the lower experienced latency than by FAPs. Fig. 3.6 illustrates the distribution of cumulative rewards for different clusters in each epoch, where the number of clusters is

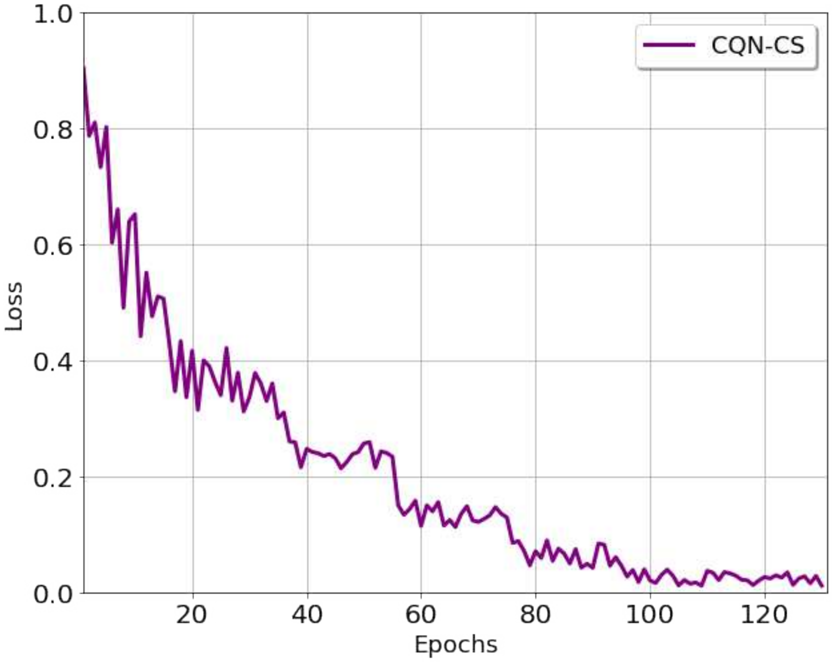


Figure 3.4: The convergence of the proposed CQN-CS framework.

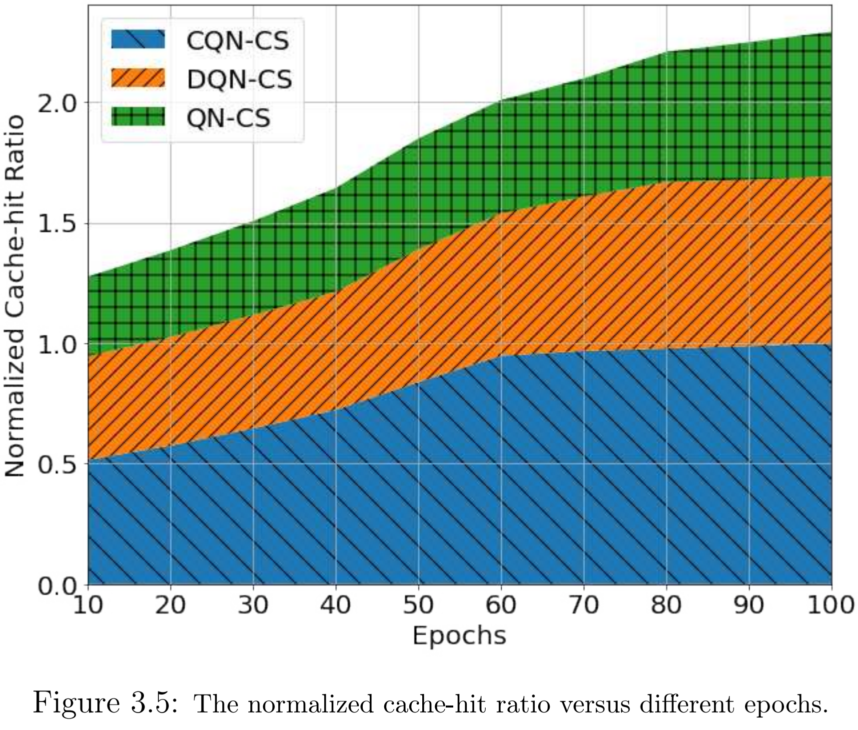


Figure 3.5: The normalized cache-hit ratio versus different epochs.

equal to the number of UAVs (i.e., 10). Then, we run the program for 10 iterations. As it can be seen from Fig. 3.6, the cumulative rewards variations of the proposed CQN-CS framework is much less than the DQN-CS and the

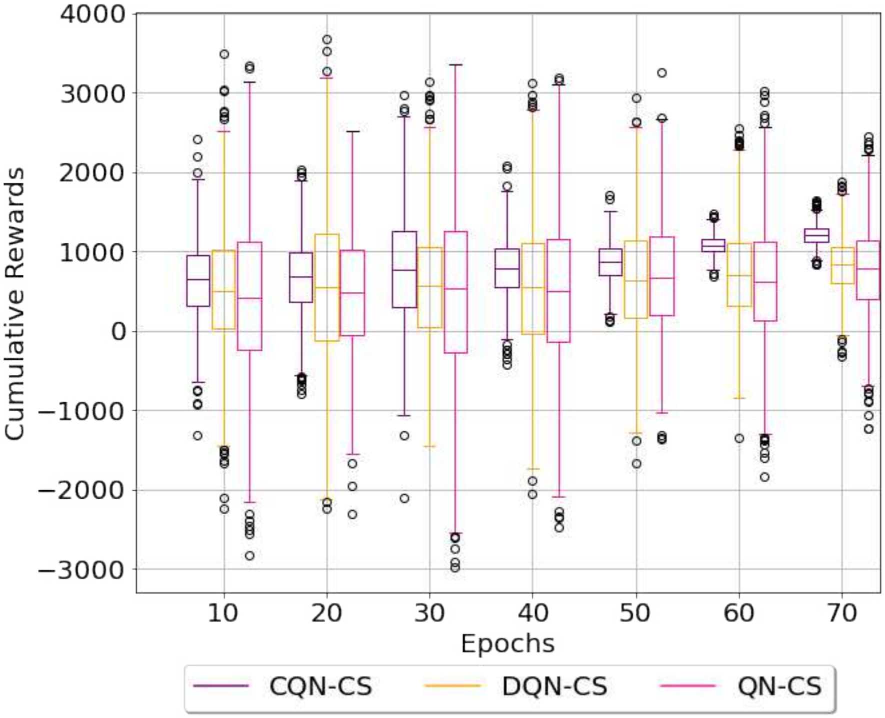


Figure 3.6: The variation of cumulative rewards versus different epochs.

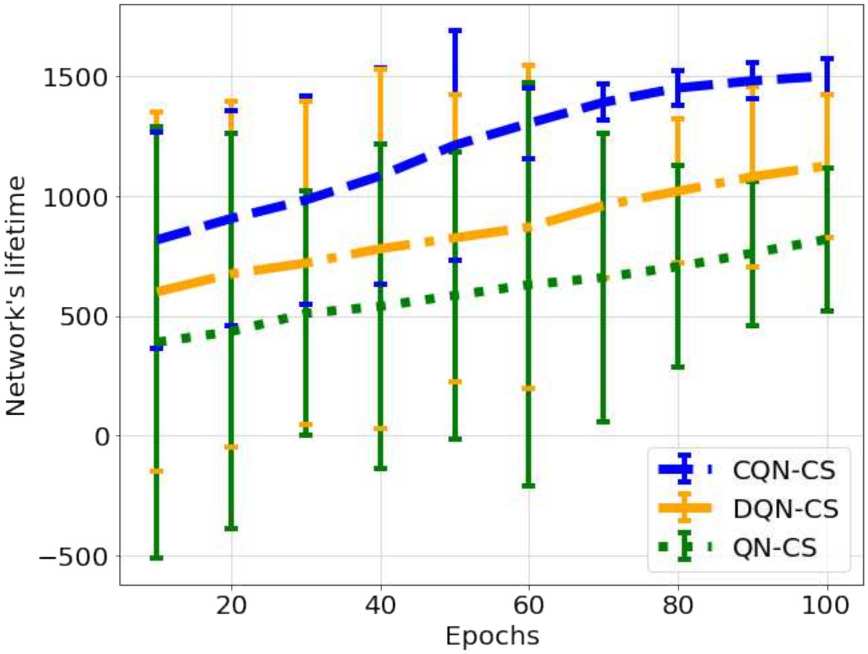


Figure 3.7: Normalized lifetime of the network in each epoch.

QN-CS approaches, which means our proposed algorithm reaches the optimum

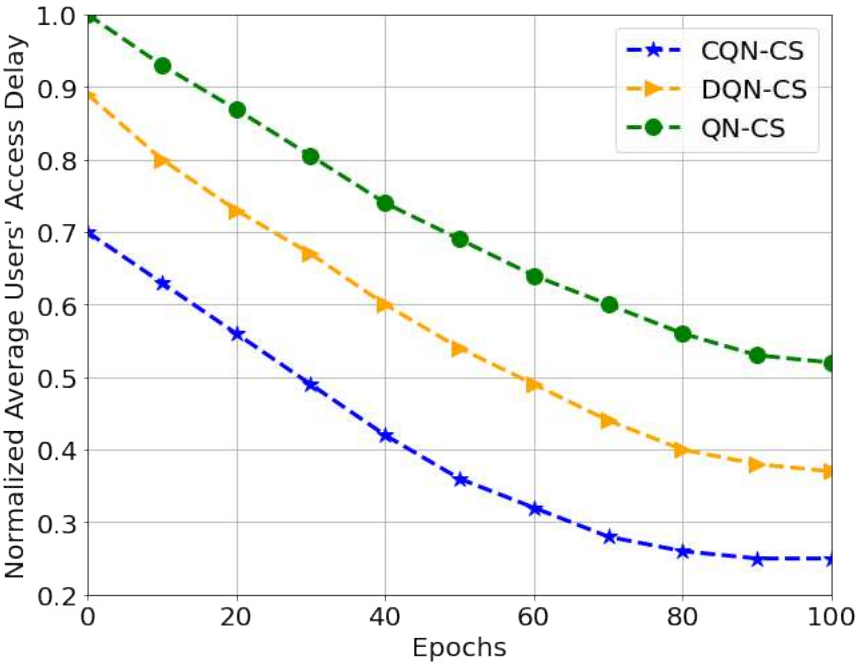


Figure 3.8: The normalized average users' access delay versus different epochs.

connection scheduling immediately.

Fig. 3.7 evaluates the performance of the proposed CQN-CS algorithm with other schemes from the aspect of the network's lifetime, depending on the energy consumption of UAVs. Note that there are three parameters which have a great impact on the energy consumption of UAVs; (*i*) The number of requests they served; (*ii*) The distance between the requested ground user and the UAV, and; (*iii*) The probability of the existence of LoS link between them. By considering all these metrics in our proposed scheme, UAVs are involved in such communications that consume less energy as much as possible, which leads to expanding the network's lifetime.

Fig. 3.8 compares the normalized average delay that all ground users in the network experience through the proposed CQN-CS, the DQN-CS, and the QN-CS frameworks. Note that the users' access delay depends on the availability of the requested content in the cache of the responsible FAP and/or UAV, otherwise, the corresponding content must be provided by the cloud server, leading to more delay. Taking into account that the location of the ground user has a great impact on the distance between the responsible caching node and the ground user, followed by the channel condition, different ground users may experience a wide range of latency. As it can be seen from Fig. 3.8, ground users during their movement and by considering the unforeseen conditions of the UAV-based femtocaching network, learn how to manage their requests by optimal caching nodes to experience lower latency.

In Fig. 3.9, we evaluate the handover rate in the proposed CQN-CS framework along with other baselines versus the number of epochs. Handover rate, denoted by $\mathcal{HR}(t)$, indicates the probability of handover, occurring between FAP f_i and the ground user GU_j at time slot t, which is obtained as

$$\mathcal{HR}(t) = \frac{P_{th}}{\mathcal{HO}_{i,j}(t)}, \tag{48}$$

where the high value of $\mathcal{HO}_{i,j}(t)$ means that the ground user connects to the close FAP instead of the far one, leading to a decrease in the handover rate. In a dynamic femtocaching network, however, ground users may become close or farther away from the nearest FAP during their movements, which is considered in our proposed CQN-CS framework, denoted by $\Delta_{i,j}(t)$. To illustrate the handover rate improvement of our proposed CQN-CS framework, we compare the handover rate, with the case that $\Delta_{i,j}(t)$ is disregarded, which is named CQN-CS2 framework. Fig. 3.9 illustrates the superiority of our proposed CQN-CS framework in terms of the average handover rate.

Finally, we compare the performance of the proposed CQN-CS framework with the DQN-CS and QN-CS baselines from the aspect of the average energy consumption of UAVs in Fig. 3.10. Using the fact that the coverage areas of UAVs are much more expanded than FAPs, all users that have access FAPs can be managed by UAVs, as well. Despite FAPs, that are unlimited energy caching nodes, serving through UAVs inherently decreases the lifetime of UAVs' battery. Consequently, it is essential to manage ground users by FAPs, especially in such cases that ground users have access to at least one FAP to extend the lifetime of the network. As it can be seen from Fig. 3.10, the average normalized energy consumption of UAVs in our proposed CQN-CS framework is lower than the other schemes. In addition, the variation of the energy consumption of UAVs is negligible in comparison to the two baselines.

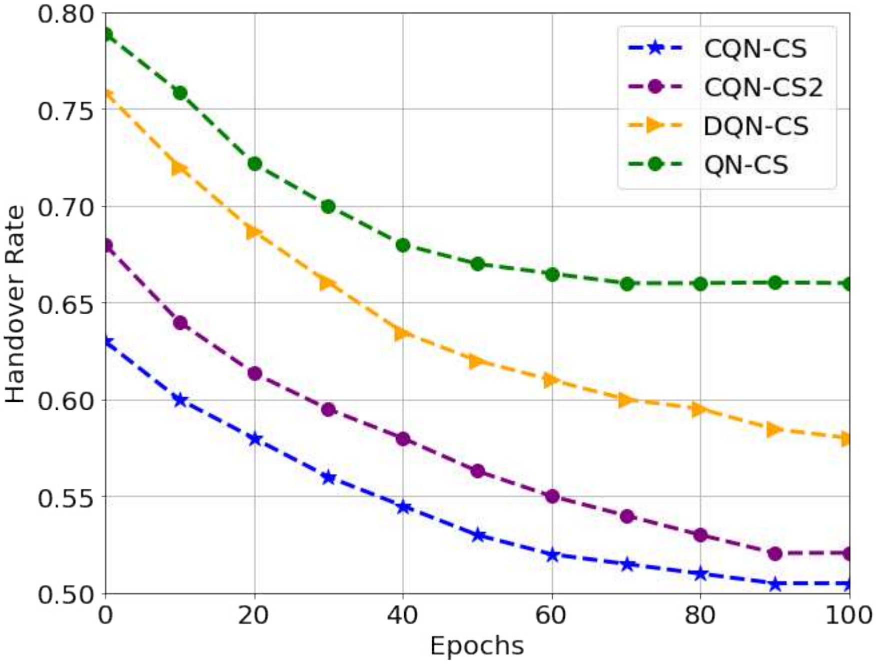


Figure 3.9: Handover rates versus different epochs.

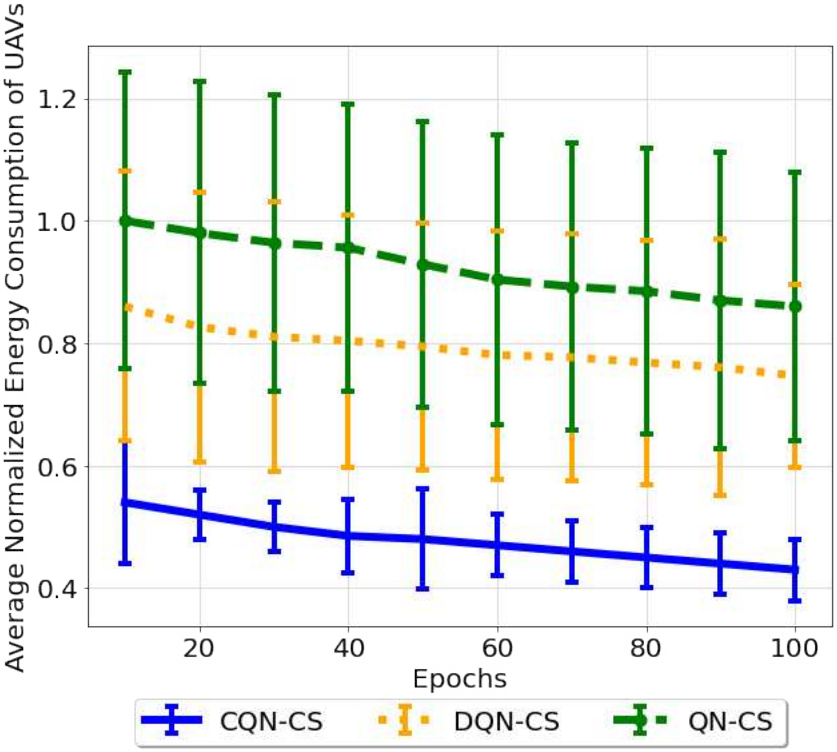


Figure 3.10: The normalized average energy consumption of UAVs versus different epochs.

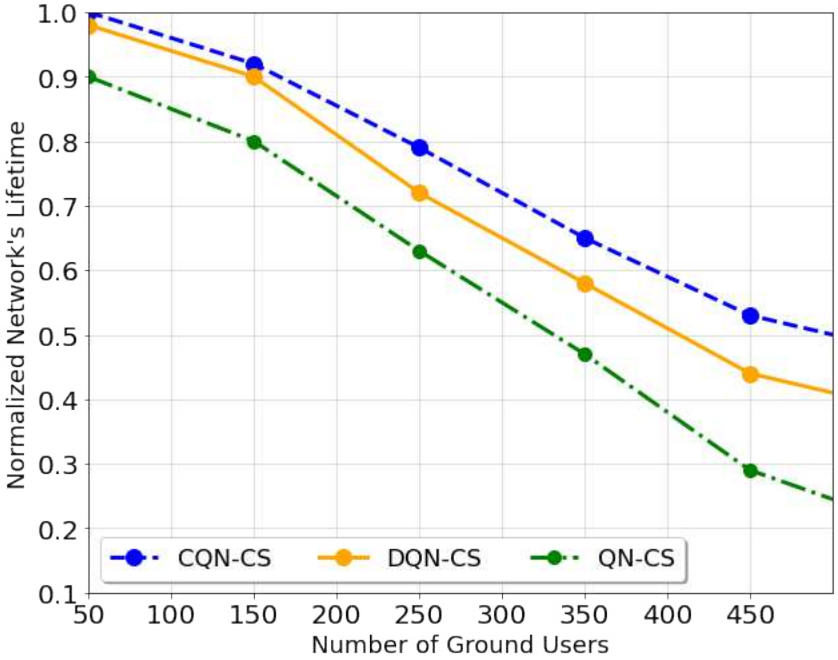


Figure 3.11: Normalized network's lifetime versus number of ground users.

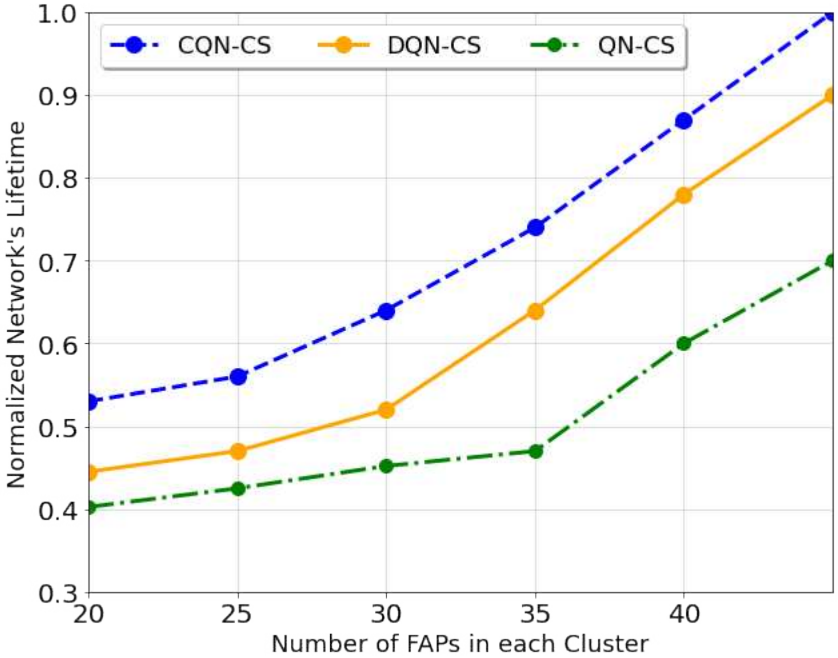


Figure 3.12: Normalized network's lifetime versus the number of FAPs in each cluster.

Moreover, we evaluate the effects of the number of ground users, the number of FAPs in each cluster, and the total number of content in the network on the performance of the UAV-based femtocaching network in Figs. 3.11-3.13. Prior research studies [114, 123] considered a small-scale wireless network with a limited number of ground users and multimedia content and illustrated the impact of the number of ground users and caching nodes on the network's performance. As it can be seen from [114, 123], considering a large number of ground users and multimedia content in wireless networks lead to a considerable increase in the number of distinct requests. In this case, providing high QoS and QoE communication links through the network is more challenging in comparison with a small-scale wireless network. Note that if a UAV-based femtocaching framework can perform effectively in an ultra-dense wireless network, it will definitely perform well in small-scale wireless networks. For this reason, we consider a sufficiently large number of ground users and multimedia content. According to the results shown in Fig. 3.11, increasing the number of ground users leads to an increase in the number of requests in each time slot. Since the network's lifetime depends on the energy consumption of UAVs, managing more requests by UAVs considerably reduces the network's lifetime. Fig. 3.12 illustrates the impact of the number of FAPs in each cluster on the network's lifetime. Increasing the number of FAPs in each cluster decreases the distance between FAPs and ground users. Therefore, ground users experience less delay by connecting to FAPs instead of UAVs. Therefore, the majority of requests will be managed through FAPs, which increases the network's lifetime. Finally, Fig. 3.13 evaluates the normalized network's lifetime versus the total number of content in the network. Note that increasing the number of content increases the content diversity throughout the network. Consequently, the number of requests that can directly be served through caching nodes decreases. Therefore, the requested content should be provided by the cloud server, leading to consuming more energy by UAVs.

Fig. 3.14 illustrates the robustness of the proposed CQN-CS method against the Channel State Information (CSI) uncertainty. It should be noted that the state space of the proposed deep Q-Network, including users' access delay, handover, and UAV's energy consumption is prone to noise. Consequently, the CSI uncertainty and RSSI measurement errors have potential negative impacts on both the state and action spaces. To evaluate the noise robustness of the proposed

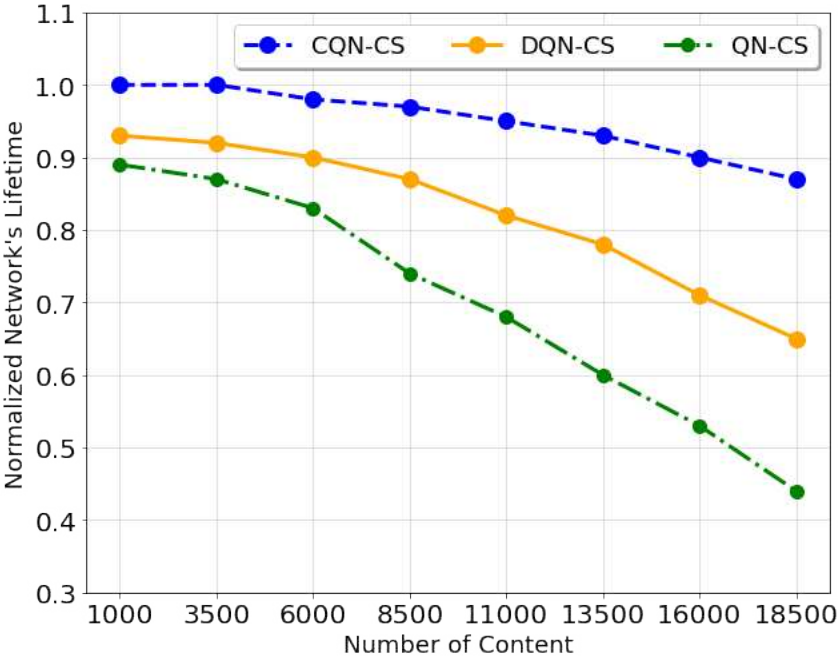


Figure 3.13: Normalized network's lifetime versus number of multimedia content.

CQN-CS, we have computed the cumulative reward (after a specific epoch where the proposed CQN-CS framework is well trained) versus different values of the noise power. By considering the fact that the common value of noise power in UAV-based femtocaching networks is in the range of $-174 \leq N_0 \leq -94$ dBm [108, 161], our proposed CQN-CS framework is robust against CSI uncertainty and RSSI measurement errors for a small value of noise (see Fig. 3.14). It is worth mentioning that in the existing literature on UAV-based femtocaching, commonly -174 dBm is used as the value of the noise power [112, 161]. It can be observed from Fig. 3.14 that the proposed CQN-CS is robust in this vicinity. On the other hand, to evaluate the performance of the proposed approach, comparison studies are performed based on the worst-case scenario (-94 dBm) and as can be seen in Figs. 3.5-3.13, our proposed method outperforms its counterparts in the worst-case scenario. To further improve the robustness of the proposed CQN-CS, robust RL models [125] can be incorporated within the CQN-CS framework, which is the focus of our ongoing research.

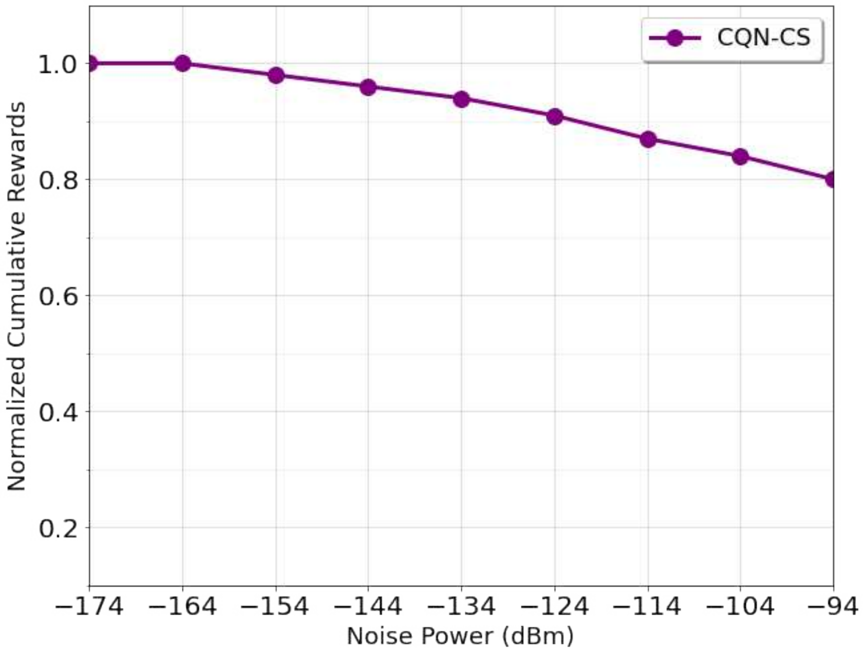


Figure 3.14: Normalized cumulative rewards versus different values of noise power (dBm).

3.2 Attention-based Popularity Prediction

In this Section, we introduce a Vision Transformer-based Edge (TEDGE) caching framework with the application to the MEC networks. The proposed TEDGE framework learns the real-time caching strategy from sequential requests of multimedia content. The main objective of several recent time-series prediction models that have been applied to the multimedia content caching is to predict the underlying patterns of the future multimedia content requests, i.e., the number of content requests using historical information. Considering the fact that the users' preferences remain unchanged for a while [57], it is sufficient to predict the potential Top-K popular content using the learned patterns from historical requests. The main focus of this study, therefore, is to predict the Top-K popular content using historical information instead of predicting the number of upcoming requests. More precisely, the TEDGE caching framework is a multi-label classification model with the aim of minimizing the difference between the actual Top-K popular content and the predicted ones. To simultaneously analyze the sequential pattern of all content, the TEDGE caching framework employs a Vision Transformer (ViT) architecture instead of using conventional Transformer models. The input of the ViT model is an image, where each pixel indicates the number of requests of each content in a specific time. In what follows, first we briefly present the MovieLens dataset, then we introduce the TEDGE framework.

3.2.1 Dataset Pre-Processing

In this study, we use MovieLens Dataset [212], which is one of the well-known movie recommendation services. In this dataset, movies with related information such as movie titles, release date, and genre are provided. Each content is requested by several users in different timestamps, where the contextual information of users such as age, gender, occupation, and their ZIP codes are also released. With the assumption that users leave a comment after watching a movie [55, 214] and in order to extract the content request pattern, commenting on a content is considered as a request. Moreover, to identify the users' location in each timestamp, ZIP codes are converted to longitude and latitude coordinates [55]. Considering the limited transmission range of caching nodes, caching nodes' locations, and users locations, the available caching nodes for serving requests of all users will be determined. Our main goal in the TEDGE caching framework is to monitor the historical requests pattern of each content to predict the Top-K popular content in an upcoming time period. Therefore, the preparation of the dataset is performed in the following four steps:

Step 1 (Request Matrix Formation): In the first step, the dataset is sorted for each content c_l, for $(1 \leq l \leq N_c)$, in the ascending order of time. Therefore, we form an $(T \times N_c)$ indicator request matrix for each hgNB, denoted by $\mathbf{R}$, where T and N_c represent the total number of timestamps and the total number of distinct content, respectively. In the request matrix, $r_{t,l} = 1$ illustrates that content c_l is requested at time t; otherwise, $r_{t,l} = 0$.

Step 2 (Time Windowing): Considering the fact that the most popular content should be cached at the storage of caching nodes during the off-peak time [126], there is no need to predict the content popularity at each timestamp.

We, therefore, define the updating time t_u (i.e., the off-peak time), as the timestamp that the storage of caching nodes is updated by the new popular content. In this case, we will have a time window with the length of $\mathcal{W}$, where $\mathcal{W}$ is associated to the time duration between two updating times, and the number of time windows is represented by $N_{\mathcal{W}} = \frac{T}{\mathcal{W}}$. Therefore, we have a $(N_{\mathcal{W}} \times N_c)$ window-based request matrix, denoted by $\mathbf{R}^{(\mathcal{W})}$, where $r^{(w)}_{t_u,l} = \sum\limits_{t=(t_u-1)\mathcal{W}+1}^{t_u\mathcal{W}} r_{t,l}$ illustrates the total number of requests of content c_l between updating time $t_u - 1$ and t_u.

Step 3 (Data Segmentation): As mentioned previously, the main target of the TEDGE caching framework is to use the historical information of content to predict the Top-K popular content in the next updating time. Given the window-based request matrix $\mathbf{R}^{(\mathcal{W})}$, the collected request pattern data is segmented via an overlapping sliding window of length l. As it can be seen from Fig. 3.15, the window-based request matrix $\mathbf{R}^{(\mathcal{W})}$ is converted into $\mathcal{D} = \{(\mathbf{X}_u, \mathbf{y}_u)\}_{u=1}^{M}$, where M represents the total number of segments. Moreover, terms $\mathbf{X}_u \in \mathbb{R}^{l\times N_c}$ and $\mathbf{y}_u \in \mathbb{R}^{N_c\times 1}$ represent the request pattern of all content before updating time t_u with the length of l, and its corresponding label, respectively. Considering the fact that there are N_c number of content through the network, and our objective is to predict the Top-K popular content, the problem at hand is a multi-label classification, where $\mathbf{y}_u$ is a vector with K ones and $y_{ul} = 1$ illustrates that content c_l would be popular at t_{u+1}. Therefore, c_l should be stored at the storage of hgNB to increase the cache-hit-ratio.

Step 4 (Data Labeling): Due to the limited storage of caching nodes, it is sufficient to identify the Top-K popular content, instead of predicting the popularity of all content at each updating time. According to the request pattern of multimedia content, we calculate the probability of requesting content c_l, for $(1 \leq l \leq N_c)$, which is obtained as follows

$$p_l^{(t,b)} = \frac{r^{(w)}_{t_u,l}}{\sum\limits_{l=1}^{N_c} r^{(w)}_{t_u,l}}. \tag{49}$$

Note that, relying on the probability of content as a single criteria for identifying

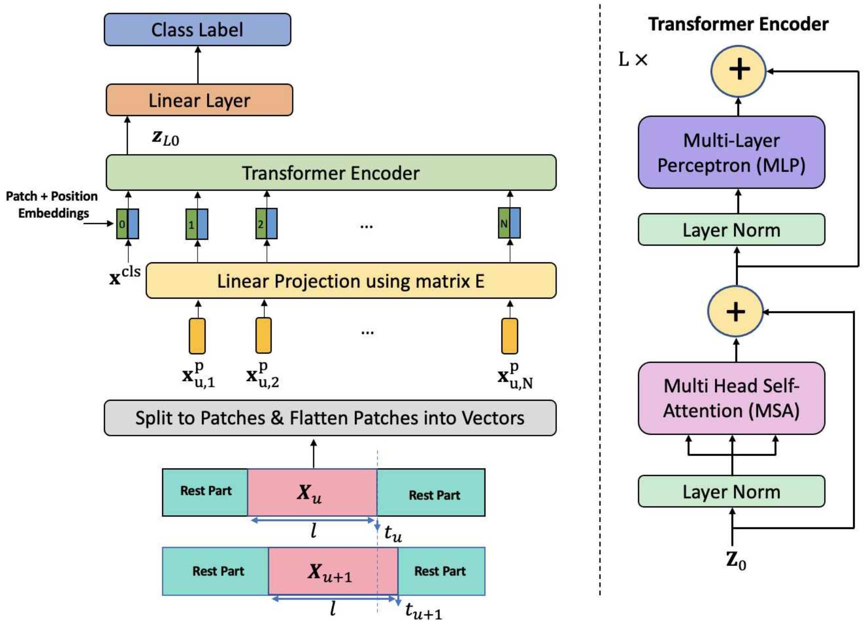


Figure 3.15: Left: Pipeline of the ViT Architecture, Right: Architecture of the Transformer Encoder.

the popularity of content has the following disadvantages: (i) Popular content with a high number of requests will be identified as the Top-K popular content for a long time, even if they are becoming unpopular, and; (ii) The popularity of new/unknown coming content (first appearance) would be predicted with a considerable delay, because the cumulative number of requests of such content is less than other content that are existing for a long time. To tackle with this issue, we use the skewness of the request pattern as another metric, which is a widely used indicator in time-series forecasting models [127]. The skewness of content c_l is denoted by ζ_l, where $\zeta_l < 0$ shows that the number of requests of content c_l increases over time. Finally, the Top-K content with the highest probability and the negative skew will be labeled as the Top-K popular content. This completes presentation of the data preparation for training the TEDGE caching framework. Next, we present the ViT architecture.

3.2.2 TEDGE Caching Framework

In this Subsection, we present the TEDGE caching framework, which is designed to predict the Top-K popular content. Generally speaking, the main characteristics of Transformers are as follows: (i) *Non-sequential:* Unlike RNN, the Transformer's attention mechanism makes it unnecessary to process data in the same order. As a result, Transformer is more parallelization than RNNs, which means it takes less time to train; (ii) *Self Attention*, indicating the similarity scores between different elements of a sequential data, and; (iii) *Positional Embeddings:* Since Transformers are non-sequential learning models, the order of information in a sequential data is missing. Therefore, Positional embeddings is introduced for recovering position information. As it can be seen from Fig. 3.15, the TEDGE caching framework consists of the following three modules: (a) Patch and Position Embeddings; (b) Transformer encoder, and; (c) MLP head, which are described as follows:

Patch and Position Embeddings: As it can be seen from Fig. 3.15, the segmented CP data $\mathbf{X}_u$ is split into N sequence of non-overlapping patches with the fixed-size of $(S \times S)$, where the total number of patches is $N = w/S^2$, with $w = l \times N_c$. After this step, each patch is flattened into a vector $\mathbf{x}^p_{u,j} \in \mathbb{R}^{S^2}$ for $(1 \leq j \leq N)$. To embed vector $\mathbf{x}^p_{u,j} \in \mathbb{R}^{S^2}$ into the model's dimension d, a linear projection $\boldsymbol{E} \in \mathbb{R}^{S^2 \times d}$ is used, which is shared among all patches, where the output of this projection is referred to as the patch embeddings. We append a learnable embedding token $\mathbf{x}^{cls}$ to the beginning of the sequence of embedded patches [128]. Finally, the position embeddings $\boldsymbol{E}^{pos} \in \mathbb{R}^{(N+1)\times d}$, is added to the patch embeddings to explicitly encode the order of the input sequence. The output of the patch and position embeddings Z_0 is given by

$$Z_0 = [\mathbf{x}^{cls}; \mathbf{x}^p_{u,1}\boldsymbol{E}; \mathbf{x}^p_{u,2}\boldsymbol{E}; \ldots; \mathbf{x}^p_{u,N}\boldsymbol{E}] + \boldsymbol{E}^{pos}. \tag{50}$$

Transformer Encoder: Given the output of the linear projection, the sequence of vectors Z_0 is fed to the transformer encoder [65]. As it can be seen from Fig. 3.15, the transformer encoder consists of L layers, with two modules, i.e., the Multihead Self-Attention (MSA) mechanism, and the MLP modules, where MLP module consists of two linear layers with Gaussian Error Linear Unit (GELU) activation function. The output of the MSA and MLP modules of layer l, for $(1 \leq l \leq L)$ are given by

$$\begin{aligned} Z_l^{'} &= MSA(LayerNorm(Z_{l-1})) + Z_{l-1}, & (51) \\ Z_l &= MLP(LayerNorm(Z_l^{'})) + Z_l^{'}, & (52) \end{aligned}$$

where a layer-normalization is used to avoid the degradation problem [197]. Finally, the output of the Transformer is

$$Z_L = [\mathbf{z}_{L0}; \mathbf{z}_{L1}; \ldots; \mathbf{z}_{LN}], \tag{53}$$

where $\mathbf{z}_{L0}$ is used for classification purposes, which is passed to a Linear Layer (LL), i.e.,

$$\mathbf{y} = LL(LayerNorm(\mathbf{z}_{L0})). \tag{54}$$

This completes the description of the Transformer autoencoder. Next, we present the description of the SA and the MSA, respectively.

1. Self-Attention (SA): The SA module [65] is used in the Transformer architecture to focus on significant parts of a given input by capturing the interaction between different vectors in $Z \in \mathbb{R}^{N \times d}$, where Z consists of N vectors, each with an embedding dimension of d. Towards this goal, three different matrices are defined, named Queries $\boldsymbol{Q}$, Keys $\boldsymbol{K}$, and Values $\boldsymbol{V}$, computed by a linear transformation as follows

$$[\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}] = ZW^{QKV}, \tag{55}$$

where $W^{QKV} \in \mathbb{R}^{d \times 3d_h}$ represents the trainable weight matrix, and d_h is the dimension of $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$. The SA block measures the pairwise similarity between each query and all keys. The output of the SA block $SA(Z) \in \mathbb{R}^{N \times d_h}$, which is the weighted sum over all values $\boldsymbol{V}$, is given by

$$SA(Z) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_h}})\boldsymbol{V}, \tag{56}$$

where term $\frac{\boldsymbol{QK}^T}{\sqrt{d_h}}$ is the scaled dot-product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ by $\sqrt{d_h}$ and softmax is used to convert the scaled similarity to the probability.

2. Multihead Self-Attention (MSA): In the MSA, the SA block is performed h times in parallel, which results in attending to information from different representation subspaces at different positions. More precisely, MSA module consists of h heads with different trainable weight matrix for each head $\{W_i^{QKV}\}_{i=1}^{h}$. After applying the SA mechanism on input Z for each head (Eqs. (55)-(56)), the outputs of h heads are concatenated into a single matrix $[SA_1(Z); SA_2(Z); \ldots; SA_h(Z)] \in \mathbb{R}^{N \times h.d_h}$. Finally, the output of the MSA module is given by

$$MSA(Z) = [SA_1(Z); SA_2(Z); \ldots; SA_h(Z)]W^{MSA}, \tag{57}$$

where $W^{MSA} \in \mathbb{R}^{hd_h \times d}$ and d_h is set to d/h.

3.2.3 Simulation Results

In this Subsection, we first evaluate different variants of the proposed TEDGE caching framework to obtain the best architecture through the process of trial and error. Considering the location of UE, which is obtained from the ZIP code and followed by Reference [63], six caching nodes are employed in different areas, where the classification accuracy is averaged over all caching nodes. In all experiments, the one-dimensional time-series content's request data is converted to a sequential set of images, which is known as the Gramian Angular Field (GAF) technique [129]. Utilizing GAF method, not only the temporal characteristics of the data is preserved, but also the temporal correlations of data are included. In Back Propagation (BP) training, Adam optimizer is employed, where the weight decay and betas are set to 0.001 and $(0.9, 0.999)$. The size of the input image, the size of input patches, and the batch size are (25×25), (5×5), and 256, respectively. Finally, we use binary cross-entropy as the loss function for our multi-label classification problem. According to the results in Table 3.3, increasing the model dimension from 32 to 128 (Model 1 to Model 3), and also the number of MLP layers from 1 to 3 (Models 3, 5, and

Table 3.3: Variants of the TEDGE caching framework.

Model ID	**Layers**	**Model dimension d**	**MLP layers**	**MLP size**	**Heads**	**Params**	**Accuracy**	**Loss**
1	1	32	1	256	8	887,140	91.09 %	0.2023
2	1	64	1	256	8	1,822,055	91.70 %	0.1888
3	1	128	1	256	8	3,913,063	92.61 %	0.1645
4	2	128	1	256	8	4,506,983	91.68 %	0.1961
5	1	128	2	256	8	3,978,852	93.19 %	0.1495
6	1	128	3	256	6	3,912,807	92.89 %	0.1600
7	1	128	3	256	8	4,044,644	**93.72** %	0.1391
8	1	128	3	256	10	4,176,487	92.61 %	0.1650
9	1	128	1	512	8	7,215,719	91.48 %	0.2060

6) increase the classification accuracy, while increasing the number of trainable parameters. Moreover, we evaluate the effect of MLP size on the classification accuracy. As it can be seen from Table 3.3, increasing the MLP size from 256 to 512 (Model 3 and Model 9) decreases the classification accuracy. Similarly, there is no improvement in the classification accuracy by increasing the number of transformer layers (see Models 3 and 4). Furthermore, considering Model 6 to Model 8, the best number of heads in this architecture is equal to 8. Note that we also evaluated the effect of window length on the classification accuracy, while no improvement has been achieved by changing the window length.

Finally, we compare the performance of the proposed TEDGE caching framework with five state-of-the-art caching schemes on the Movielens dataset, including LRU, LFU, PopCaching [214], LSTM-C [55], and the TRansformer (TR) caching, which is an upgraded version of the attention-based neural network in Reference [215]. While the attention-based model in Reference [215] is used for predicting the request pattern of online content, we adopt it to predict the Top-K popular ones. Fig. 3.16 compares the performance of the proposed TEDGE scheme with other baselines mentioned above from the aspect of the cache-hit ratio, when the DNN models reach the steady state. In the content caching context, cache-hit-ratio is a widely used metric, illustrating the ratio of requests served by caching nodes versus total requests. Considering the Zipf distribution for the content popularity profile, we set the storage capacity of caching nodes to 10% of the total content [131]. As shown in Fig. 3.16, the optimal strategy [55] is a caching scheme, where all requests are served through caching nodes, which cannot be obtained in reality. According to the results in Fig. 3.16, the proposed TEDGE caching framework obtains the highest cache-hit-ratio in comparison to its state-of-the-art counterparts.

3.3 ViT-CAT Architecture

Despite the fact that the TEDGE framework achieves a high level of classification accuracy, it is intricate and demands a significant number of parameters in order to achieve such accuracy. In this Section, we present a parallel Vision Transformers with Cross Attention (ViT-CAT) Fusion architecture to predict

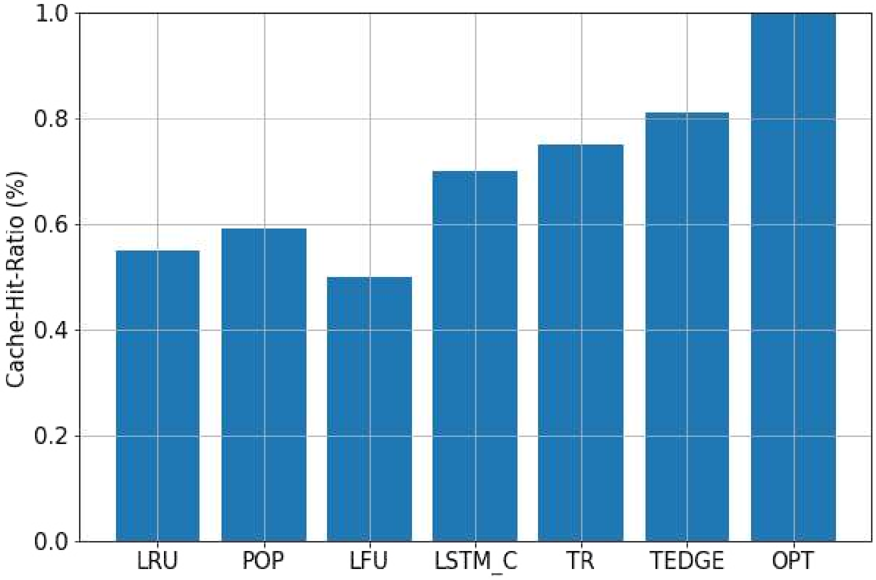


Figure 3.16: A comparison with state-of-the-arts based on the cache-hit-ratio.

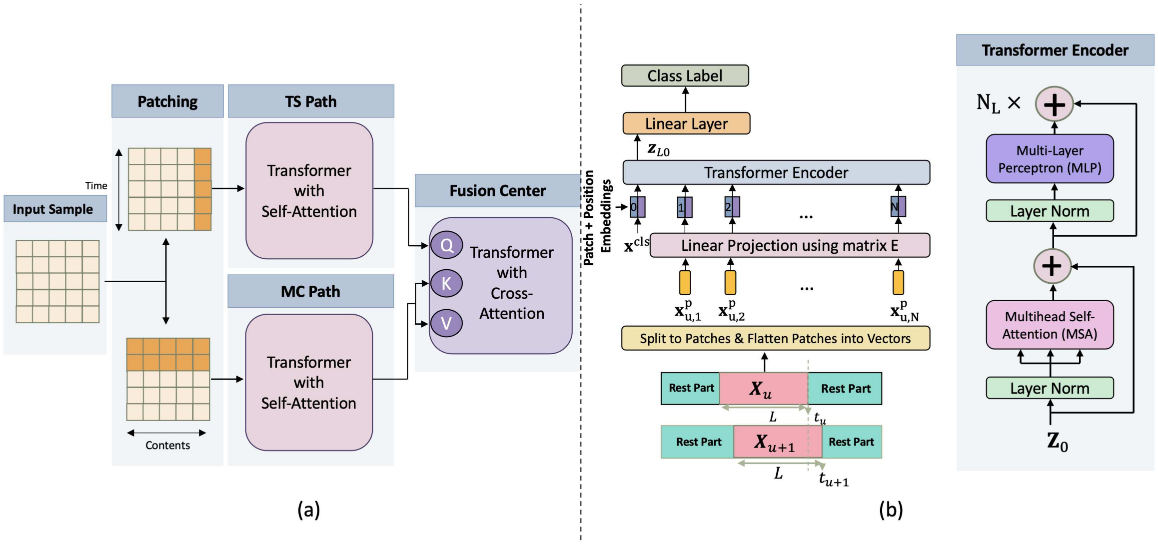


Figure 3.17: (a) Block diagram of the proposed ViT-CAT architecture, (b) Pipeline of the ViT architecture, and the Transformer encoder.

the Top-K popular contents in MEC networks. As shown in Fig. 3.17(a), the ViT-CAT architecture consists of two parallel paths, named Time-Series (TS)-path and Multi-Content (MC)-path, performed based on self-attention mechanism, followed by a Cross-Attention (CA) module as the fusion layer.

A. Patching

Generally speaking, the input of the Transformer encoder in the ViT network is a sequence of embedded patches, consisting of patch embedding and positional embedding. In this regard, the 2D input samples $\mathbf{X}_u$ is split into N non-overlapping patches, denoted by $\mathbf{X}_{u,p} = \{\mathbf{x}_{u,p}^{(i)}\}_{i=1}^{N}$. As can be seen from Fig. 3.17(a), we apply two following patching methods for TS-path and MC-path:

i) *Time-based Patching*: To capture the temporal correlation of contents, we use time-based patching for the TS-path, where the size of each patch is $S = L \times 1$. More precisely, time-based patching separately focuses on the request pattern of each content for a time sequence with a length of L, where the total number of patches is $N = (L \times N_c)/(L \times 1) = N_c$, which is the total number of contents.

ii) *Content and Time-based Patching*: In the MC-path, the main objective is to capture the dependency between all N_c contents for a short time horizon T_s, therefore, we set the size of each patch to $S = T_s \times N_c$, where the number of patches is $N = (L \times N_c)/(T_s \times N_c) = L/T_s$, with $T_s \ll L$.

B. Embedding

Each patch is flattened into a vector $\mathbf{x}_{u,j}^{p} \in \mathbb{R}^{1\times S}$ for $(1 \leq j \leq N)$. Referred to as the patch embedding, a linear projection $\boldsymbol{E} \in \mathbb{R}^{S\times d}$ is used to embed vector $\mathbf{x}_{u,j}^{p}$ into the model's dimension d. Then, a learnable embedding token $\mathbf{x}^{cls}$ is added to the beginning of the embedded patches. Finally, to encode the order of the input sequences, the position embedding $\boldsymbol{E}^{pos} \in \mathbb{R}^{(N+1)\times d}$, is appended to the patch embedding, where the final output of the patch and position embeddings, denoted by Z_0, is given by

$$Z_0 = [\mathbf{x}^{cls}; \mathbf{x}_{u,1}^{p}\boldsymbol{E}; \mathbf{x}_{u,2}^{p}\boldsymbol{E}; \ldots; \mathbf{x}_{u,N}^{p}\boldsymbol{E}] + \boldsymbol{E}^{pos}. \tag{58}$$

C. Cross-Attention (CA)

The CA module is the same as the SA block, except that the Query $\boldsymbol{Q}$, Key $\boldsymbol{K}$, and Value $\boldsymbol{V}$ are obtained from different input features as shown in Fig. 3.17(a). More precisely, to learn the mutual information between TS and MC paths, the Query $\boldsymbol{Q}$ comes from the output features of TS-path, while Key $\boldsymbol{K}$, and Value $\boldsymbol{V}$ are obtained from the output features of the MC-path. More precisely, the output of the one layer to the last of each path is given to the CA block. The CA block has three inputs, which are key, query, and value, where query comes from the output of the TS-path (one layer to the last of the TS-path), while key and value are the same and are obtained from the layer before the last layer of the CA-path. Finally, the output of the CA block is given to a fully connected layer, where the size of the output layer is the same as the number of contents and the activation function is Sigmoid as this is a multi-label classification. This completes the description of the proposed ViT-CAT architecture.

3.3.1 Simulation Results

In this Subsection, we evaluate the performance of the proposed ViT-CAT architecture through a series of experiments. Given the users' ZIP code in Movielens dataset [63], we assume there are six caching nodes, where the classification accuracy is averaged over all caching nodes. In all experiments, we use the Adam optimizer, where the learning and weight decay are set to 0.001 and 0.01, respectively, and binary cross-entropy is used as the loss function for the multi-label classification task. In Transformers, the MLP layers' activation function is ReLU, whereas their output layer's function is sigmoid.

Effectiveness of the ViT-CAT Architecture: Different variants of the proposed ViT-CAT architecture are evaluated to find the best one through trial and error. According to the results in Table 3.4, increasing the MLP size from 64 (Model 4) to 128 (Model 6), the model dimension from 25 to 50 (Model 1 to Model 2), the number of MLP layers from 1 to 2 (Models 4 and 5), the number of heads from 4 to 5 (Models 2 and 3), and the number of Transformer layers from 1 (Model 2) to 2 (Model 6) increase the classification accuracy while increasing the number of parameters.

Effect of the Fusion Layer: In this experiment, we evaluate the effect of the fusion layer in the proposed ViT-CAT architecture with other baselines, where the parallel ViT architecture in all networks is the same (Model 2), i.e., the TS and MC paths perform based on the SA mechanism. In this regard, we consider two fusion layers, i.e., the Fully Connected (FC) and the SA layers. According to the results in Table 3.5, the CA module outperforms the others,

Table 3.4: Variants of the ViT-CAT Architecture.

Model ID	**Layers**	**Model dimension d**	**MLP layers**	**MLP size**	**Heads**	**Params**	**Accuracy**
1	1	25	1	128	5	201,188	84.35 %
2	1	50	1	128	5	435,788	94.77%
3	1	50	1	128	4	415,488	80.35 %
4	1	50	1	64	5	342,793	93.49 %
5	1	50	2	64	5	435,788	94.82%
6	2	50	1	128	5	568,185	94.84 %

Table 3.5: Classification accuracy using different fusion networks.

Model	**CA**	**FC**	**SA**
Accuracy	94.77%	92.58 %	79.93 %
Parameters	435,788	417,171	400,788

since it captures the mutual information between two parallel networks.

Performance Comparisons: Finally, we compare the performance of the proposed ViT-CAT architecture in terms of the cache-hit ratio with other state-of-the-art caching strategies, including LSTM-C [55], TRansformer (TR) [215], ViT architecture [5], and some statistical approaches, such as Least Recently Used (LRU), Least Frequently Used (LFU), PopCaching [214]. We consider two versions of the ViT architecture with the lowest complexity, called ViT-LC, and the highest accuracy (ViT-HA). With the assumption that the storage capacity of caching nodes is 10% of total contents [131], a high cache-hit ratio illustrates that a large number of users' requests are managed through the caching nodes. As shown in Fig. 3.18, the optimal strategy, which cannot be attained in a real scenario, is one in which all requests are handled by caching nodes. The cache-hit ratio of the ViT-CAT (0.815) is almost the same as the ViT-HA (0.807), since the higher classification accuracy, results in a higher cache-hit ratio, while the complexity of the ViT-CAT architecture is about 8 times lower than the ViT-HA. More precisely, the highest accuracy of the ViT-CAT model is 94.84% with $568,185$ number of parameters, while in a single ViT network, the best performance occurs with 93.72% accuracy and $4,044,644$ number of parameters.

3.4 Self-Supervised Popularity Prediction

A DNN-based popularity prediction model needs to have the ability to understand the historical request patterns of content, including their temporal and spatial correlations. Existing state-of-the-art time-series DNN models capture the latter by simultaneously inputting the sequential request patterns of multiple contents to the network, considerably increasing the size of the input sample.

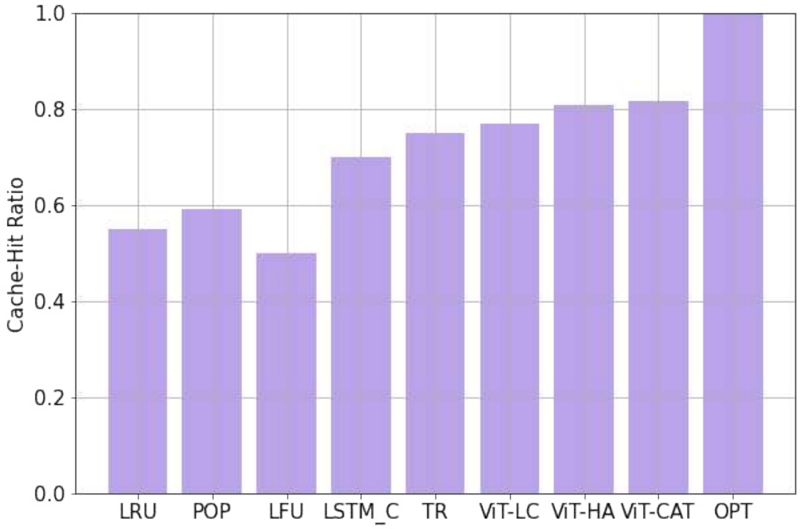


Figure 3.18: Comparison with state-of-the-arts based on the cache-hit ratio.

This motivates us to address this challenge by proposing a DNN-based popularity prediction framework based on the idea of contrasting input samples against each other, designed for the UAV-aided MEC networks. Referred to as the Contrastive Learning-based Survival Analysis (CLSA), the proposed architecture consists of a self-supervised Contrastive Learning (CL) model, where the temporal information of sequential requests is learned using a Long Short Term Memory (LSTM) network as the encoder of the CL architecture. Followed by a Survival Analysis (SA) network, the output of the proposed CLSA architecture is probabilities for each content's future popularity, which are then sorted in descending order to identify the Top-K popular contents.

3.4.1 Dataset Pre-Processing

MovieLens dataset [212], as one of the most well-known movie recommendation services that provide users' contextual information, is used in this study to evaluate the proposed CLSA architecture. MovieLens was generated on October 17, 2016, including the movie rates that 943 users gave to 1682 movies between September 19, 1997, and April 22, 1998. As shown in Fig. 3.19(a), *u.user* document provided in the MovieLens dataset, contains users' contextual information, including gender, age, occupation, and ZIP code. The ZIP codes are converted to latitude and longitude coordinates to extract users' locations during their requests. Another document provided in this dataset is

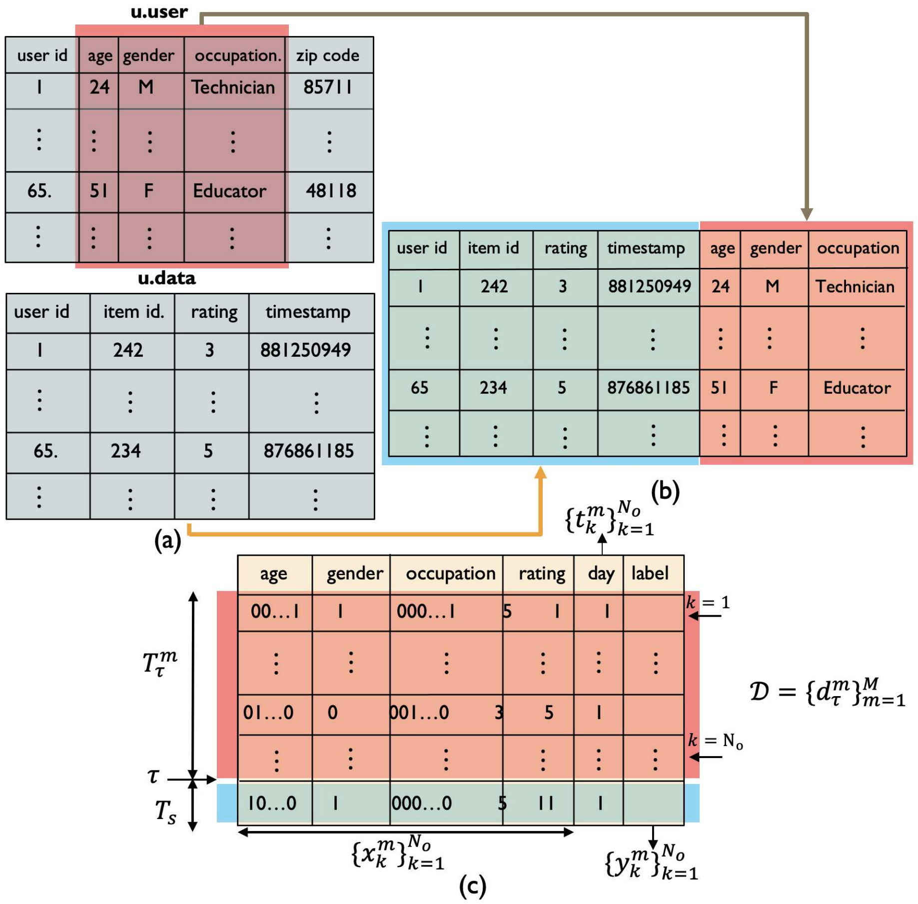
Table 3.6: List of Notations.

Notation	**Description**	**Notation**	**Description**
u_s	Unmanned Aerial Vehicle (UAV)	N_u	Number of UAVs
f_i	Femto Access Point (FAP)	N_f	Number of FAPs
u_l	User	U	Number of users
c_m	Content	M	Number of contents
T_τ^m	Observational window at time τ	T_s	Study window
N_o	Number of requests in T_τ^m	d_τ^m	Time-series observational data of c_m
x_τ^m	Users' information and rating	h_k^m	Latent representation of x_k^m
t_k^m	Rating time	$h_k^{m,(SH)}$	Shuffled encoded sample
y_k^m	Label of content	$h_k^{m,(MA)}$	Masked encoded sample
$d_m^{(MA)}$	Masked version of d_τ^m	$L_{cl}^{(MA)}$	Masking loss function
$d_m^{(SH)}$	Shuffled version of d_τ^m	$L_{cl}^{(SH)}$	Shuffling loss function
L_{cl}	Contrastive learning loss	L_{re}	Reconstruction loss
L_{sa}	Survival analysis loss	T_{total}	Total time window
$\overline{x}_k^m$	Decoded version of x_k^m	p_t^m	Probability of getting popular

u.data, including user ID, item ID (content ID), the rate the user gave to the corresponding content, and the timestamp that the user watched and rated the content. A summary of the notations used hereinafter is provided in Table 3.6.

As shown in Fig. 3.19, the following steps are performed to adopt the Movielens dataset with the CLSA architecture: (*i*) The *u.data* and *u.user* documents are concatenated on "user ID" (see Fig. 3.19(b)) and the ZIP code is dropped from the concatenated dataset; (*ii*) The concatenated dataset in Fig. 3.19(b) is sorted by "item ID", and "timestamp"; (*iii*) The "user ID" and "item ID" are dropped from the concatenated dataset, since these columns are not informative; (*iv*) The categorical features including gender, age, and occupation are encoded using a one-hot encoder; (*v*) The timestamp is discretized with a resolution of one day since the storage of edge devices should be updated during the off-peak times (i.e., midnight) [126]. Then, the discretized timestamp, named "day", is replaced with the timestamp column; and, (*vi*) Finally, a column named "label" is added to the concatenated dataset, indicating the content popularity (popular or unpopular), which will be described shortly.

We define an observational window for each content c_m, for $(1 \leq m \leq M)$ having the length of T_τ^m at time τ, where the request pattern of contents within this range are studied to predict their popularity in the future, i.e., the study window, denoted by T_s. Following the Reference [201], we consider the same number of requests N_o in the observational window for all the contents, where zero padding is used for contents with less number of requests. Given the request patterns of contents over the observational window at time τ, the popularity of contents is predicted within the time window $[\tau, \tau + T_s]$. Without loss of generality, since the storage capacity of edge devices should be updated each day, we assume that the length of the study window is $T_s = 1$. Given M number of multimedia contents and U number of users, the dataset $\mathcal{D} = \{d_\tau^m\}_{m=1}^M$ is created, where $d_\tau^m = \{(x_k^m, t_k^m, y_k^m)\}_{k=1}^{N_o}$ is the time-series observational data for content c_m. Term $\{x_k^m\}_{k=1}^{N_o}$ includes the contextual information of users requesting content c_m and the rating that these users gave to content c_m (see Fig. 3.19(c)). Moreover, rating time is represented by $\{t_k^m\}_{k=1}^{N_o}$. Finally, term $y_k^m \in \{0, 1\}$ is associated with the content popularity, which is 1 if content c_m gets popular during the study window T_s, otherwise, $y_k^m = 0$. Consequently,



user id	age	gender	occupation.	zip code
1	24	M	Technician	85711
⋮	⋮	⋮	⋮	⋮
65.	51	F	Educator	48118
⋮	⋮	⋮	⋮	⋮

user id	item id.	rating	timestamp
1	242	3	881250949
⋮	⋮	⋮	⋮
65.	234	5	876861185
⋮	⋮	⋮	⋮

user id	item id	rating	timestamp	age	gender	occupation
1	242	3	881250949	24	M	Technician
⋮	⋮	⋮	⋮	⋮	⋮	⋮
65	234	5	876861185	51	F	Educator
⋮	⋮	⋮	⋮	⋮	⋮	⋮

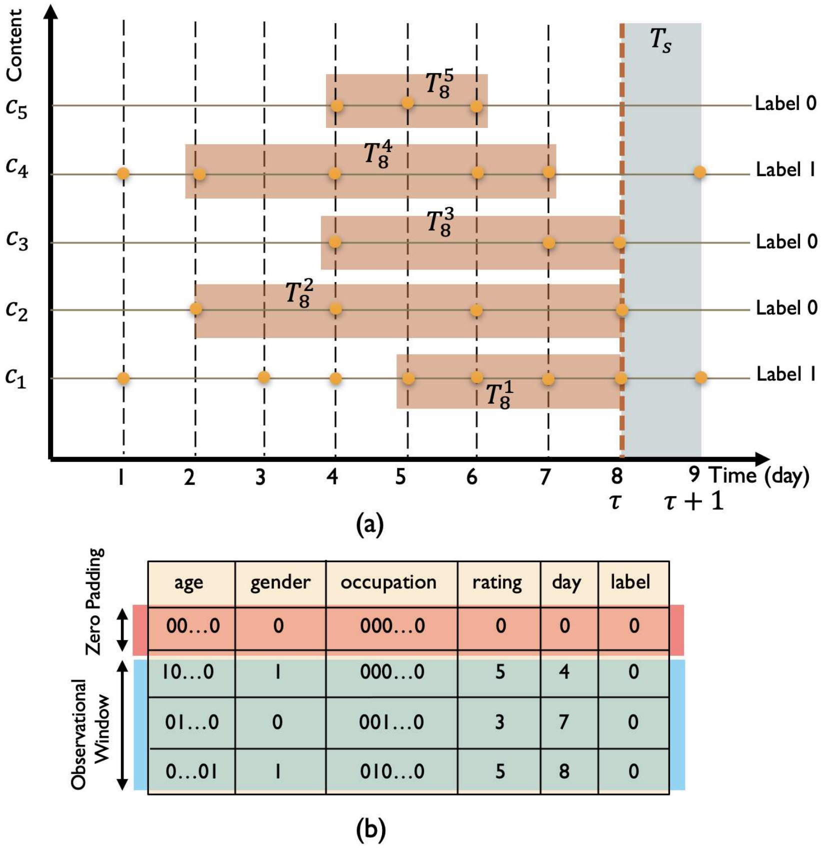
age	gender	occupation	rating	day	label
00…1	1	000…1	5 1	1	
⋮	⋮	⋮	⋮	⋮	⋮
01…0	0	001…0	3 5	1	
⋮	⋮	⋮	⋮	⋮	⋮
10…0	1	000…0	5 11	1	

Figure 3.19: (a) A typical sample of the Movielens dataset, (b) the concatenated dataset, and (c) the adopted version of the Movielens dataset used for the CLSA architecture.

given d_τ^m, the output of the proposed CLSA model predicts whether or not content c_m will be popular during the study window T_s.

To clarify the benefits of the proposed strategy, an illustrative example of constructing input samples and their labels is depicted in Fig. 3.20. In this illustrative toy-example, it is assumed that there are 5 contents $\{c_m\}_{m=1}^{5}$ across the network, and the goal is to update the storage of edge devices at time $\tau + T_s = 9$ using the historical requests of contents up to time $\tau = 8$ with $T_s = 1$. For instance, the observational window $\{T_\tau^m\}_{m=1}^{5}$ would be T_8^1 for content c_1. Moreover, $N_o = 4$ number of requests for each content are investigated during the observational window. As shown in Fig. 3.20(a), the following cases can occur:

1. In the first case, content c_1 is requested at time $\tau = 8$. Ending at time $\tau = 8$, the observational window T_8^1 is determined in such a way that the



age	gender	occupation	rating	day	label
00…0	0	000…0	0	0	0
10…0	1	000…0	5	4	0
01…0	0	001…0	3	7	0
0…01	1	010…0	5	8	0

Figure 3.20: (a) An illustrative example of request patterns of contents to construct input samples and their labels. (b) Zero padding technique is used for contents with the number of requests less than N_o.

number of requests during this observational window is $N_o = 4$, where $T_8^1 = 3$. Therefore, the requests within $T_8^1 = 3$ ending at $\tau = 8$ are considered to evaluate the popularity of content c_1 at time $\tau + T_s = 9$. Consequently, $\{x_k^1\}_{k=1}^4$ includes the contextual information of users requesting content c_1 and their ratings. As shown in Fig. 3.20(a), since content c_1 is requested at time 9, its label is $\{y_k^1\}_{k=1}^4 = 1$.

2. In the second case (i.e., content c_2), there is a request at time $\tau = 8$ and the observational window is $T_8^2 = 6$ for having $N_o = 4$ number of requests within this range. However, since this content is not requested at time 9, its label is $\{y_k^2\}_{k=1}^4 = 0$.
3. In the third case (i.e., content c_3), the content is requested at $\tau = 8$, but the number of existing requests from the beginning to $\tau = 8$ is less than $N_o = 4$. Therefore, we use zero padding to create the same length of

input samples, shown in Fig 3.20(b), where zero pad will be added at the beginning of the observational window.

4. In the fourth case, content c_4 is not requested at time $\tau = 8$, but there is at least $N_o = 4$ number of requests before $\tau = 8$. Therefore, the last $N_o = 4$ of requests within the observational window T_o^4 are used as the input sample.
5. In the fifth case, content c_5 is not requested at time $\tau = 8$, and the number of requests from the beginning to time $\tau = 8$ is less than $N_o = 4$. Similar to the case 3, the zero padding technique is used to create the input sample.
6. Finally, if there is a content with zero requests up to time $\tau = 8$, this content will be removed from the study.

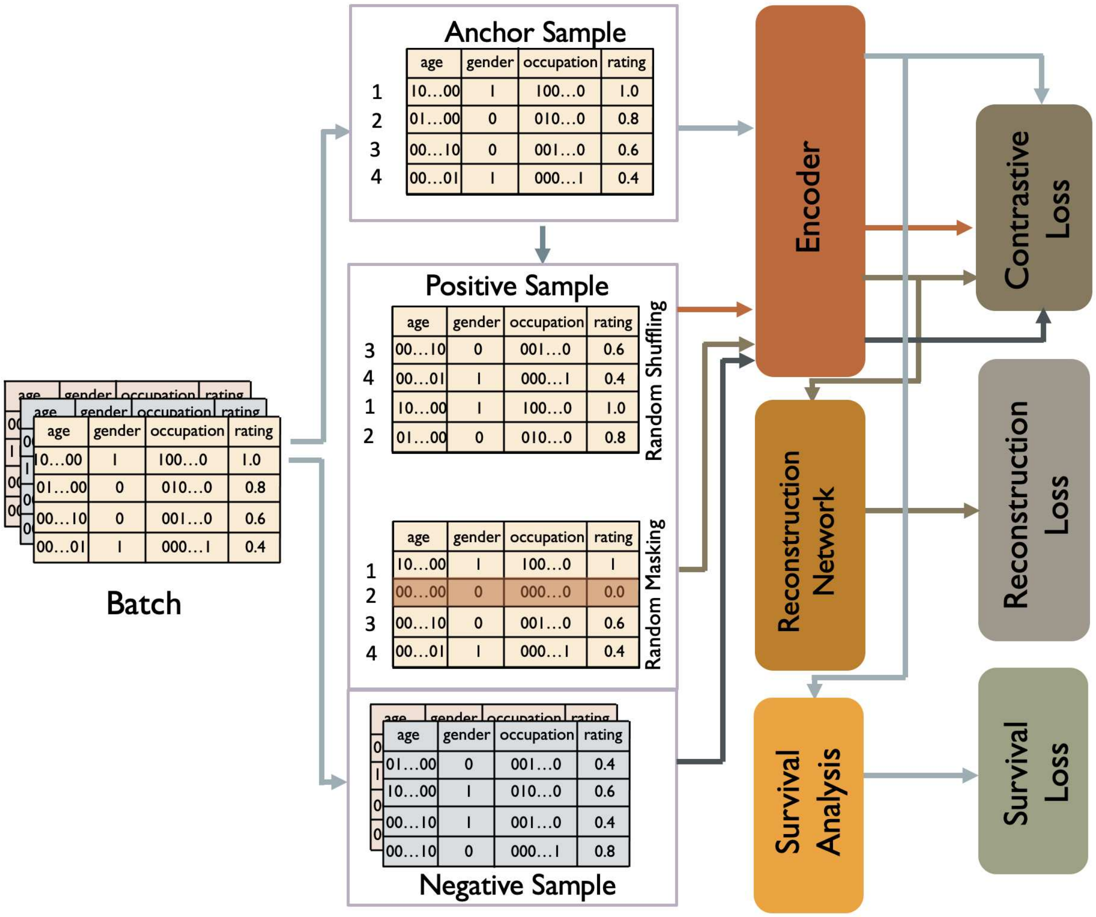
This completes the problem description and system modeling, next, we present the proposed CLSA framework.

3.4.2 Proposed CLSA Framework

In this Subsection, we present the constituent components of the proposed CLSA framework. As can be seen from Fig. 3.21, the CLSA architecture consists of three modules, i.e., CL, RN, and SA models. The CA network is utilized to capture the spatial and temporal correlations of input samples by converting the longitudinal input data $x_k^m \in \{d_\tau^m\}_{m=1}^M$ to the latent representations. The RN block is then used to decode the latent representation. Given a meaningful latent representation, the SA model is used to predict the probability that content c_m will become popular. In the following, first, we briefly introduce the CL block, then, we explain RN and the SA models.

Contrastive Learning (CL) Block

CL model, as one of the widely used self-supervised learning paradigms in computer vision, has recently been applied to tabular and longitudinal datasets. The fundamental principle of the CL model is to acquire the ability to differentiate between pairs of examples that are similar, also known as positive pairs,



	age	gender	occupation	rating
	10…00	1	100…0	1.0
	01…00	0	010…0	0.8
	00…10	0	001…0	0.6
	00…01	1	000…1	0.4

	age	gender	occupation	rating
1	10…00	1	100…0	1.0
2	01…00	0	010…0	0.8
3	00…10	0	001…0	0.6
4	00…01	1	000…1	0.4

	age	gender	occupation	rating
3	00…10	0	001…0	0.6
4	00…01	1	000…1	0.4
1	10…00	1	100…0	1.0
2	01…00	0	010…0	0.8

	age	gender	occupation	rating
1	10…00	1	100…0	1
2	00…00	0	000…0	0.0
3	00…10	0	001…0	0.6
4	00…01	1	000…1	0.4

age	gender	occupation	rating
01…00	0	001…0	0.4
10…00	1	010…0	0.6
00…10	1	001…0	0.4
00…10	0	000…1	0.8

Figure 3.21: The overall perspective of the CLSA architecture.

and those that are dissimilar, or negative pairs. To achieve the objective of learning these representations effectively, it is essential to have both positive and negative examples. Positive examples refer to pairs of data that share specific features or characteristics, which should be grouped together. Conversely, negative examples denote pairs that do not possess those shared features and should be separated. Without positive examples, the model would be unable to learn the shared characteristics between pairs of data, which is necessary to group them together. Similarly, without negative examples, the model would be unable to learn the features that differentiate dissimilar pairs of data. Hence, having both positive and negative examples allows the model to learn how to distinguish between similar and dissimilar pairs of data. As shown in Fig. 3.21, given a sample as the input of the learning model, called an anchor sample, we need to select negative and positive samples for that. The negative samples consist of all the other samples within the batch, as they differ from the anchor sample. However, we must generate positive samples that are similar to the anchor sample. In the proposed CLSA framework, we use the following two augmentation methods to create positive samples, named shuffling and masking techniques (in what follows, for simplicity, we drop subscript τ wherever there is no ambiguity):

- **Masking:** Given the longitudinal data d_m of content c_m, several users' information $\{x_k^m\}_{k=1}^{N_o}$ is randomly masked to generate a positive sample for content c_m, denoted by $d_m^{(MA)}$. Other historical request patterns of contents in the batch are represented by negative samples.
- **Shuffling:** In this type of data augmentation, the positive sample is generated by randomly shuffling the time order of users' information requesting content c_m. The shuffled sample is denoted by $d_m^{(SH)}$.

It is important to note that the sample data is comprised of a series of N_o requests that are sequential, and their order is significant as each request corresponds to a row in the sample data. As depicted in Fig. 3.21, shuffling involves randomly altering the order of requests, whereas masking removes an entire row from the sequence. By utilizing these two methods, we can produce two new samples that are similar to the original anchor sample. As an example, we examine four requests (rows) in the anchor sample shown in Fig. 3.21, with their order represented as 1, 2, 3, and 4. Fig. Fig. 3.21 illustrates the random shuffling of the rows, resulting in the order 3, 4, 1, 2. Additionally, for random masking, each row's information is randomly obscured by zero values. As an example, we have randomly selected row 2 to be masked with zero values. Furthermore, in order to clarify the presence of multiple 0 and 1 in the illustrative tables shown in Fig. Fig. 3.21, it is important to mention that we employed a one-hot encoder for the age and occupation columns, which generates several zeros and ones in the table. Additionally, we normalized the rating column, ensuring that the final rating falls within the range of 0 to 1.

Given the augmented and negative samples, a shared encoder is utilized to convert the longitudinal input data to a meaningful latent representation. This representation places contents with similar request patterns closer together and dissimilar contents farther apart. To preserve the temporal correlation of request patterns, an LSTM architecture is used as the encoder, where the latent representation of x_k^m at time t_k^m, denoted by h_k^m, is given by

$$h_k^m = LSTM(x_k^m, h_{k-1}^m), \qquad k = 1, \ldots, N_o. \tag{59}$$

In addition to the anchor sample d_m, the shuffled and masked version of d_m are given as the input of the encoder to efficiently learn the latent representation of d_m. The shuffled and masked encoded samples are represented by $h_k^{m,(SH)}$, and $h_k^{m,(MA)}$, respectively.

Finally, we utilize the masked and shuffled CL loss functions, denoted by $L_{cl}^{(MA)}$, and $L_{cl}^{(SH)}$, respectively, where the shuffled/masked learned representation is considered the positive sample, and other representations in the batch are considered the negative ones, as follows

$$L_{cl}^{(MA)} = -\sum_{m=1}^{M} \log \frac{\exp\left(h_{N_o}^m (h_{N_o}^{m,(MA)})^T\right)}{\sum\limits_{j=1, j\neq m}^{M} \exp\left(h_{N_o}^m (h_{N_o}^{j,(MA)})^T\right)}, \tag{60}$$

$$L_{cl}^{(SH)} = -\sum_{m=1}^{M} \log \frac{\exp\left(h_{N_o}^m (h_{N_o}^{m,(SH)})^T\right)}{\sum\limits_{j=1, j\neq m}^{M} \exp\left(h_{N_o}^m (h_{N_o}^{j,(SH)})^T\right)}, \tag{61}$$

where $(.)^T$ is the transpose function, and the total CL loss is $L_{cl} = L_{cl}^{(MA)} + L_{cl}^{(SH)}$. In general, the CL model is capable of distinguishing between different samples by transforming them into a latent representation. This transformation ensures that similar samples are grouped close to each other, while dissimilar samples are placed further apart in the latent space. The key to achieving this objective lies in the loss function used by the model, particularly in minimizing the expressions defined in Eqs. (60) and (61). These loss functions have a crucial role in guiding the model to minimize the distance or maximize the similarity between positive pairs, which consist of similar samples (i.e., anchor and positive samples). Conversely, the loss functions also aim to maximize the distance or minimize the similarity between negative pairs, which consist of dissimilar samples (i.e., anchor and negative samples). To minimize the loss function, it is important for the numerator to have a small value, while the denominator should be large. Eqs (60) and (61) provide further insight into the loss functions. The numerator in these equations indicates the similarity between the anchor sample and the positive sample, i.e., $h_{N_o}^{m}$ and $h_{N_o}^{m,(MA)}$, respectively, in Eq. (60) and $h_{N_o}^{m}$ and $h_{N_o}^{m,(SH)}$, respectively in Eq. (61), and minimizing this term helps to achieve the desired similarity. On the other hand, the denominator represents the similarity between the anchor sample $h_{N_o}^{m}$ and the remaining masked samples in the current batch, i.e., $h_{N_o}^{j,(MA)}$, where $j = \{1, \ldots, N\}$ and $j \neq m$ in Eq. (60) and maximizing this term ensures that distinct values are assigned to different samples in the latent representation. With the same argument, the denominator of Eq. (61) attempts to maximize the distance between $h_{N_o}^{m}$ and $h_{N_o}^{j,(SH)}$ where $j = \{1, \ldots, N\}$ and $j \neq m$. This completes the presentation of the CL component of the CLSA architecture. Next, we present the reconstruction network.

Reconstruction Network (RN) Block

To learn a better latent representation, the masked version of x_k^m is regenerated from the encoded masked input sample. Following Reference [132] and to preserve the temporal information of $\{x_k^m\}_{k=1}^{N_o}$, the Time-LSTM2 is utilized as the decoder, where the regenerated sample is given by

$$\overline{x}_k^m = Time - LSTM2(H_k^m), \tag{62}$$

where $\overline{x}_k^m$ is the decoded version of x_k^m, and $H_k^m = [(h_1^{m,(MA)}, t_2^m - t_1^m), (h_2^{m,(MA)}, t_3^m - t_2^m), \ldots, (h_k^{m,(MA)}, t_{k+1}^m - t_k^m)]$ with $k \in \{1, \ldots, N_o\}$. Term $h_k^{m,(MA)}$ is the masked encoded sample at time t_k^m, and $t_k^m - t_{k-1}^m$ is the time difference between two consecutive requests of content m. By minimizing the difference between the original input sample x_k^m and the decoded one $\overline{x}_k^m$, our goal is to provide a better latent representation. Therefore, the RN loss, denoted by L_{re}, is given by

$$L_{rn} = \sum_{k=1}^{N_o} ||\overline{x}_k^m - x_k^m||^2. \tag{63}$$

Note that Time-LSTM2 [132] is an enhanced version of the LSTM architecture that incorporates a temporal attention mechanism. In the proposed CLSA

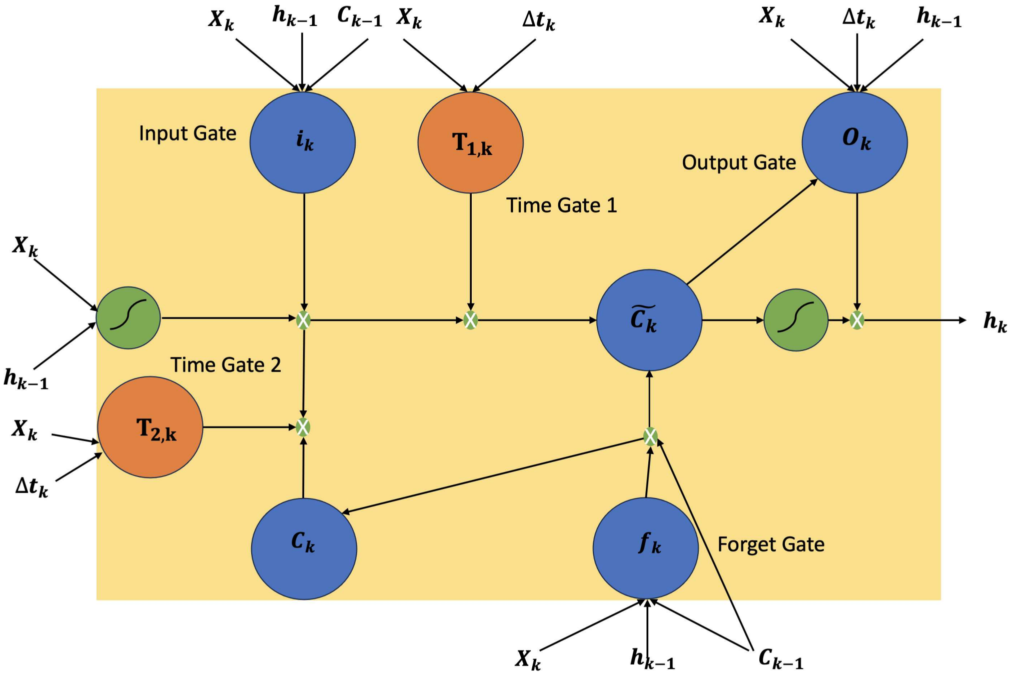


Figure 3.22: Time-LSTM2 architecture.

framework, we recognized that the time intervals between requests contain valuable temporal information regarding historical content patterns. As a result, we employed the Time-LSTM2 model as the decoder for reconstructing the masked positive sample, instead of using a simple LSTM network, to have a more accurate latent representation. Fig. 3.22 illustrates the architecture of Time-LSTM2, which consists of five gates: Input, Output, Forget, and two Time gates. As shown in the Fig. 3.22, i_k, o_k, and f_k correspond to the input, output, and forget gates of the k_{th} request, respectively. Additionally, x_k, h_k, and c_k represent the input feature, hidden output, and cell activation vectors, respectively. Furthermore, $T_{1,k}$ and $T_{2,k}$ denote two time gates. $T_{1,k}$ controls the influence of the previous request on the current popularity status, while $T_{2,k}$ captures time intervals to account for users' long-term interests in future predictions. Lastly, Δt_k represents the time interval between two requests. For more detailed information, please refer to Reference [132].

Survival Analysis (SA) Block

We utilize a SA model to capture a meaningful relationship between the longitudinal and contextual information of users requesting content c_m and the probability of getting popular in the future. To this end, we use an MLP network, where the input of this model is the latent representation $\{h^m_{N_o}\}^M_{m=1}$, and the output is the probability that content c_m gets popular at time t, denoted by $p^m_t = P(t|h^m_{N_o})$, during the total time window, denoted by T_{total}, where $T_{total} = T_s + \max\limits_{1 \leq m \leq M} T^m_o$. We use the Sigmoid function as the activation function of the output layer of the MLP network to compute p^m_t. Consequently, the estimated Cumulative Incidence Function (CIF) for content c_m at time $\tau + T_s$ is calculated as follows

$$\overline{F}^m(\tau + T_s|h^m_{N_o}) = \frac{\sum\limits_{\tau \leq t \leq \tau + T_s} P(t|h^m_{N_o})}{1 - \sum\limits_{t \leq t^m_{N_o}} P(t|h^m_{N_o})}, \tag{64}$$

where $\overline{F}^m(\tau + T_s|h^m_{N_o})$ indicates the probability that content m has gained popularity up to time $\tau + T_s$. Finally, the negative log-likelihood is used as the loss function for the SA network, given by

$$L_{sa} = -\sum_{m=1}^{M} \log\left(1 - \overline{F}^m(\tau^m|h^m_{T_m})\right). \tag{65}$$

Finally, $\{p^m_t\}^M_{m=1}$ are sorted in the descending order to identify the Top-K popular contents.

3.4.3 Simulation Results

Experimental Configurations

To investigate the effectiveness of the proposed CLSA architecture, a UAV-aided MEC network, consisting of 4 terrestrial and 2 aerial caching nodes with 943 users and 1682 multimedia contents. Following the common assumption [131], the size of the storage capacity of caching nodes is 10% of the total contents, where the size of all multimedia contents is the same. Using the five-fold cross-validation strategy, 80% and 20% samples are used as the training dataset and the test dataset, respectively. Adam optimizer was employed to train the model, where betas are $(0.9, 0.999)$ and the weight decay is set to $1e-7$. Moreover, the l_2 regularization was set to $1e-3$ to avoid over-fitting. We trained the proposed CLSA architecture by minimizing the total loss function, denoted by L_{total}, obtained as follows

$$L_{total} = \omega_{cl} L_{cl} + \omega_{rn} L_{rn} + \omega_{sa} L_{sa}, \tag{66}$$

where ω_{cl}, ω_{rn}, and ω_{sa} represent the weight of CL, RN, and SA blocks, respectively, where the summation of them is one. In the following, the details of each learning block are described:

- **Encoder:** An LSTM network is used as the encoder, with Rectified Linear Unit (ReLU) activation and sigmoid as the recurrent activation function, where the output size of this block is denoted by D_{E}. Then, the output is given to an MLP with three layers, with the size of $\alpha_l D_{\mathrm{FI}}$, where $l \in \{1, 2, 3\}$. Term D_{FI} represents the feature dimension of the input sample, which is 28 in this work, and the hyperparameter α_l, with $l \in \{1, 2, 3\}$ is set to $\alpha_1 = 1, \alpha_2 = 3$, and $\alpha_3 = 5$.
- **Decoder:** There is a decoder in the RN block, performing based on the Time-LSTM2, where the recurrent activation function is sigmoid and the general activation is Tanh. The output size of the Time-LSTM2 network is denoted by D_{D}.
- **MLP Network:** Using batch-normalization technique, this network consists of three dense layers with the same size of D_{M} and the Exponential Linear Unit (ELU) activation function. There are more 2 dense layers after that, where the size of each layer is T_{total}. The activation function of the last layer is softmax, while the rest is ReLU.

Effectiveness of the CLSA Architecture

To evaluate the performance of the proposed CLSA architecture, we first consider different variants of the CLSA model by changing different hyperparameters, such as the batch size, D_{E}, D_{D}, D_{M}, and the learning rate, denoted by lr. According to the information provided in Table 3.7, five different models are defined. Moreover, we investigate the effect of the number of requests studied in the observational window, denoted by N_o, on the classification accuracy. As it can be seen from Table 3.7, decreasing the batch size from 512 to 256 (Model 1 and 2) improves the classification accuracy over all folds with different N_o. Comparing Models 2 and 3, it can be seen that reducing the encoder and decoder dimensions from 64 to 32 decreases the classification accuracy. Similarly, reducing D_{M} from 128 to 32 results in degrading the accuracy (Model 2 and Model 4). Finally, by comparing Model 1 and Model 5, it is evident that $lr = 1e - 3$ outperforms $lr = 1e - 4$.

Moreover, we investigate the effect of N_o on the classification accuracy over 5 models, where each fold is iteratively run with different random seed values. According to the results provided in Table 3.8, increasing the number of requests studied over an observational window provides more information about the behavior of users' interests in the past, improving the classification accuracy. Accordingly, from the aspect of classification accuracy, it can be seen that Model 2 with $N_o = 20$ outperforms other variants. For this reason, we have selected this model to conduct further research. Table 3.9 demonstrates the precision, recall, and F1-score for Model 2 with $N_o = 20$ over 5 folds and their average values. According to the information provided in Table 3.9, the high value of the aforementioned parameters illustrates the effectiveness of the CLSA architecture.

Fig. 3.23 represents the confusion matrix of the Model 2 with $N_o = 20$. It should be noted that a challenging issue encountered in the Movielens dataset pertains to imbalanced data, which arises due to a vast proportion of multimedia contents being unpopular. To address this problem, we implemented the random oversampling technique to increase the number of popular samples. As depicted in Fig. 3.23, there is a misclassification rate of 2.84% for unpopular contents (labeled as 0) being incorrectly classified as popular contents, which leads to the wastage of storage capacity on edge devices. Similarly, a misclassification rate of 2.06% for popular contents being classified as unpopular contents can result in failure to place highly requested contents on edge devices.

It is important to note that oversampling [133, 134] is a widely accepted technique in machine learning to address class imbalance, where one class has significantly fewer samples compared to the other. In our case, the class representing label 1 (popular contents) had considerably fewer samples than the class representing label 0 (unpopular contents). The purpose of oversampling was to mitigate this imbalance and improve the performance of the learning model by providing more representative training data for the minority class. To clarify, oversampling does not involve content-related data manipulation or resampling of specific instances. It involves generating synthetic samples based on existing samples from the minority class. These synthetic samples are created by Synthetic Minority Over-sampling Technique (SMOTE) [134] to generate new synthetic samples that resemble the popular content. The detailed process of the SMOTE method is as follows:

1. *Identifying minority class instances:* The first step of SMOTE is to identify the popular contents. These are the samples that are less prevalent in the dataset compared to the majority class.
2. *Selecting nearest neighbors:* For each popular content, SMOTE selects a defined number of its nearest neighbors from the same class. The number of neighbors is determined by the user-defined parameter k.
3. *Creating synthetic samples:* SMOTE then creates synthetic samples by interpolating between the feature vectors of the specific popular content and its selected neighbors. It does this by randomly selecting a neighbor and computing the difference between their feature vectors.
4. *Generating synthetic samples:* Once the difference between the feature vectors is calculated, SMOTE multiplies this difference by a random value between 0 and 1. This scaling factor controls the amount of synthetic data to be generated.
5. *Adding synthetic samples:* Finally, the synthetic samples are added to the original dataset, resulting in an augmented dataset with a more balanced class distribution. The process of selecting nearest neighbors, calculating the feature differences, and generating synthetic samples is repeated until the proposed CLSA framework receives sufficient exposure to the popular content (minority class), preventing bias towards the unpopular content (majority class).

Table 3.7: Variants of the CLSA architecture.

Model ID	**Batch Size**	$D_{\mathbf{E}}$	$D_{\mathbf{D}}$	$D_{\mathbf{M}}$	lr
1	512	64	64	128	$1e-3$
2	256	64	64	128	$1e-3$
3	256	32	32	128	$1e-3$
4	256	64	64	32	$1e-3$
5	256	64	64	128	$1e-4$

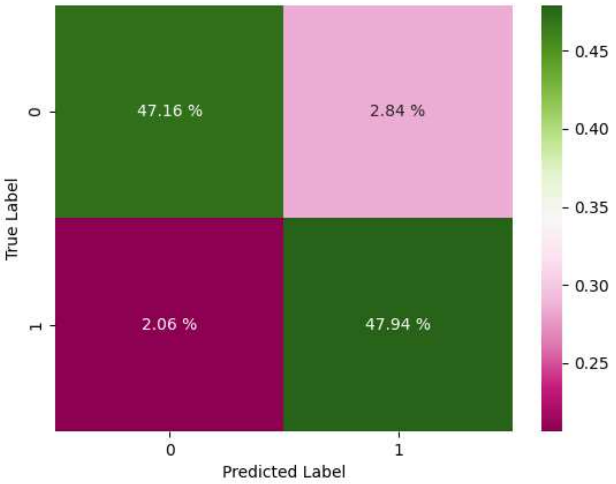


Figure 3.23: Confusion matrix of the proposed CLSA architecture (Model 2, $N_o = 20$).

Moreover, we utilize the T-distributed Stochastic Neighbor Embedding (TSNE) method [141] to evaluate the efficiency of the CL block in generating latent representations for discriminating between popular and unpopular contents. For instance, in Fig. 3.24, the latent representation of a test set from one of the 5 fold cross-validation experiments is employed to illustrate the embedded space of popular and unpopular contents. As shown in Fig.3.24, the embedded space of popular and unpopular contents are distinguishable, indicating that the CL network has been effectively trained.

Table 3.8: 5 fold cross-validation accuracy ± standard deviation for different variants of the proposed CLSA architecture using different window sizes (10, 15, and 20 days).

N_o	**Model ID**	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
10	**1**	0.850 ± 0.026	0.836 ± 0.037	0.837 ± 0.024	0.875 ± 0.022	0.831 ± 0.050	0.846 ± 0.035
	2	0.892 ± 0.031	0.909 ± 0.011	0.885 ± 0.031	0.905 ± 0.016	0.892 ± 0.021	0.896 ± 0.024
	3	0.853 ± 0.023	0.829 ± 0.030	0.826 ± 0.025	0.848 ± 0.036	0.815 ± 0.029	0.834 ± 0.031
	4	0.864 ± 0.018	0.826 ± 0.012	0.835 ± 0.015	0.849 ± 0.018	0.844 ± 0.031	0.843 ± 0.023
	5	0.866 ± 0.012	0.831 ± 0.015	0.834 ± 0.010	0.857 ± 0.012	0.859 ± 0.030	0.849 ± 0.022
15	**1**	0.890 ± 0.031	0.868 ± 0.032	0.843 ± 0.052	0.861 ± 0.027	0.861 ± 0.050	0.865 ± 0.041
	2	0.949 ± 0.019	0.940 ± 0.007	0.934 ± 0.010	0.937 ± 0.005	0.932 ± 0.012	0.938 ± 0.012
	3	0.861 ± 0.022	0.836 ± 0.024	0.842 ± 0.031	0.861 ± 0.018	0.843 ± 0.033	0.849 ± 0.027
	4	0.867 ± 0.026	0.832 ± 0.025	0.835 ± 0.024	0.868 ± 0.023	0.850 ± 0.029	0.850 ± 0.029
	5	0.864 ± 0.009	0.826 ± 0.010	0.823 ± 0.012	0.850 ± 0.021	0.870 ± 0.019	0.847 ± 0.024
20	**1**	0.878 ± 0.037	0.881 ± 0.041	0.859 ± 0.050	0.868 ± 0.058	0.855 ± 0.023	0.868 ± 0.043
	2	0.959 ± 0.003	0.942 ± 0.006	0.941 ± 0.006	0.946 ± 0.002	0.965 ± 0.003	**0.951 ± 0.010**
	3	0.866 ± 0.033	0.875 ± 0.032	0.876 ± 0.019	0.854 ± 0.024	0.845 ± 0.032	0.863 ± 0.029
	4	0.858 ± 0.016	0.845 ± 0.039	0.834 ± 0.013	0.869 ± 0.034	0.829 ± 0.065	0.847 ± 0.039
	5	0.871 ± 0.009	0.828 ± 0.013	0.830 ± 0.022	0.859 ± 0.009	0.878 ± 0.006	0.853 ± 0.024

Table 3.9: Precision, recall, and F1-score for two classes (i.e., popular (class 1) and unpopular (class 0)) for Model 2 with $N_o = 20$ using 5 fold cross-validation.

	Class	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
Precision	**0**	0.973 ± 0.012	0.934 ± 0.019	0.911 ± 0.005	0.928 ± 0.005	0.956 ± 0.010	0.941 ± 0.024
	1	0.946 ± 0.009	0.938 ± 0.019	0.976 ± 0.011	0.951 ± 0.006	0.973 ± 0.005	0.957 ± 0.018
Recall	**0**	0.945 ± 0.010	0.937 ± 0.022	0.978 ± 0.010	0.952 ± 0.006	0.974 ± 0.006	0.957 ± 0.019
	1	0.974 ± 0.012	0.934 ± 0.021	0.904 ± 0.006	0.926 ± 0.006	0.956 ± 0.010	0.939 ± 0.027
F1-score	**0**	0.959 ± 0.004	0.935 ± 0.004	0.943 ± 0.006	0.940 ± 0.005	0.965 ± 0.002	0.948 ± 0.012
	1	0.960 ± 0.003	0.935 ± 0.003	0.939 ± 0.003	0.938 ± 0.005	0.964 ± 0.003	0.947 ± 0.013

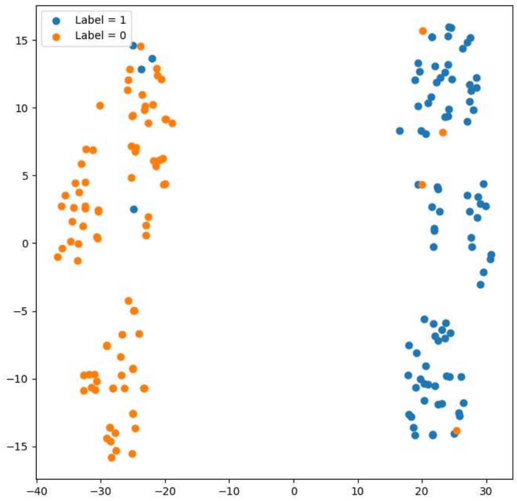


Figure 3.24: The embedded space of the latent representation of popular/unpopular contents using the TSNE technique.

Ablation Study

To further investigate the effectiveness of the proposed CLSA architecture, we conduct an ablation study, where different variants of the CLSA model are introduced in Table 3.10. We evaluate the significance of each block in the CLSA architecture on classification accuracy by analyzing Models $L1$ to $L7$. For instance, the results presented in Table 3.10 demonstrate that the CL block alone (Model $L1$) achieves an accuracy of approximately 85%, underscoring the importance of the other two blocks in attaining higher performance (Model $L7$ with about 95% accuracy). Consequently, it is evident that the presence of all three blocks in the CLSA architecture is of paramount importance, where the optimal loss weights of each block are represented in Model $L7$. It should be noted that the block weights are determined using the grid search method, where we define a grid of possible weight values for each block. Then, we train and evaluate the network using each combination of weights from the grid. The combination that yields the best performance, in terms of average accuracy, is selected as the optimal set of weights. This allows the network to learn the optimal combination of weights that maximizes the overall accuracy.

Table 3.10: The accuracy of the proposed CLSA architecture using different loss weights.

Model ID	ω_{cl}	ω_{rn}	ω_{sa}	**Average Accuracy ± STD**
$L1$	1.0	0.0	0.0	0.851 ± 0.026
$L2$	0.0	1.0	0.0	0.850 ± 0.022
$L3$	0.0	0.0	1.0	0.903 ± 0.029
$L4$	0.5	0.5	0.0	0.859 ± 0.034
$L5$	0.0	0.5	0.5	0.888 ± 0.025
$L6$	0.5	0.0	0.5	0.936 ± 0.013
$L7$	0.3	0.2	0.5	$\mathbf{0.951 \pm 0.010}$

Comparison of Available Datasets

In what follows, the impact of the dataset on the classification accuracy of the proposed CLSA framework is evaluated. The characteristics of recently available datasets are compared in Table 3.11 based on the availability of certain information, including (i) The timing of user requests, referring to information about when users made their requests; (ii) Contextual details, i.e., users' information such as age, gender, and occupation, which provides additional context about the users, and; (iii) Content ratings, indicating the users' interest in specific items, helping to understand their preferences. To the best of our knowledge, the only dataset that contains all the necessary user contextual information is the Movielens 100K dataset. Additionally, a newer version of Movielens, called Movielens 1M, provides the same information. Movielens 1M covers a longer time frame, specifically from $2000-04-25$ to $2003-02-28$, spanning a total of $1,039$ days. For Movielens 1M dataset, we set the batch size as 256, and $D_{\text{E}} = 64$, $D_{\text{D}} = 64$, $D_{\text{M}} = 128$ with $lr = 1e-3$. Moreover, the MLP used after the encoder consists of one layer, with the size of D_{FI}, and the size of the MLP network used as the survival network is T_{total}. Considering the aforementioned hyperparameters, the number of parameters of the proposed CLSA framework is $165,036$ and the classification accuracy is 95.42%. Despite the fact that the classification accuracy is only slightly higher compared to Movilens 100K, there is a significant reduction in the network's complexity, approximately four times less, which is a noteworthy improvement.

Performance Comparisons

We compare the performance of the proposed CLSA architecture with the following state-of-the-art:

- **TEDGE Scheme** [5], which is based on a simple ViT architecture acting as a multi-label classification model with the aim of predicting the Top-K popular contents in the upcoming time. To capture the spatial correlation of contents, 2D images of historical requests pattern of contents were created, where the number of columns and rows of this image were corresponding to the number of contents and the number of historical requests for each content, respectively. The size of the input sample, therefore, significantly increases to capture as much spatial correlation as possible.
- **Multiple-model Transformer-based Edge Caching (MTEC)** [9] consists of two parallel multi-channel Transformer networks with a dense layer as the fusion layer. Similarly, the output of the model is Top-K popular contents, while it first predicted the request patterns of contents in the future. The input sample is 1D historical requests patterns of contents. To capture the spatial correlation of contents, the multi-channel Transformer networks were employed, where the sequential request pattern of each content is given to a channel of the Transformer model.
- **ViT-CAT** [6], consisting of two parallel ViT networks with different patching techniques, with a cross attention mechanism as the fusion layer. The input and output of the network are similar to the TEDGE caching scheme.
- **Self-supervised Contrastive Learning Popularity Prediction (CoPo)** [7], attempting to learn the latent representation of contents to classify them as popular and unpopular contents without any labeling.
- **Deep Learning-based Content Caching (DLCC)** [142] used CNN to predict the popularity of contents and used the RL model for the content placement phase.

Table 3.11: Comparison of the information provided by different datasets.

Dataset	**Date**	**Time**	**User's Information**	**Rating**
Movielens 100K [135]	September 1997 to April 1998	✓	✓(Age, Gender, Occupation)	✓
Movielens 1M [136]	April 2000 to February 2003	✓	✓(Age, Gender, Occupation)	✓
Movielens 25M [137]	January 1995 to November 2019	✓	-	✓
Jester [138]	April 1999 to May 2003	-	-	✓
Book-Crossing [139]	August 2004 to September 2004	-	Age	✓
Last.FM [140]	March 2005 to May 2007	✓	-	✓

We compare the proposed CLSA framework with the aforementioned baselines from the classification accuracy perspective, learning efficiency, and complexity. When assessing the learning efficiency and complexity of deep learning models, the primary goal is to quantify the resources required for training and inference. Various metrics, such as time and memory usage, are commonly used to measure learning efficiency. To evaluate the learning efficiency of the proposed CLSA framework, we evaluate the train time, test time, and maximum allocated memory. As shown in Table 3.12, the proposed CLSA architecture outperforms other baselines, while there is no need to create large input samples to capture the spatial correlation of contents, i.e., the size of the input sample is 28×20. Moreover, it is important to note that the classification accuracy is equivalent to the popularity prediction error in our study. More precisely, in the proposed CLSA framework, the output variable, denoted as y_k^m, represents the popularity of content c_m during the study window T_s. Specifically, y_k^m takes the value of 1 if the content becomes popular, and 0 otherwise. This implies that the popularity prediction error aligns with the classification accuracy. Furthermore, we conducted a comparison between the cache-hit-ratio of our proposed CLSA framework and the ViT-CAT method, considering the number of epochs and the cache capacity. Fig. 3.25 illustrates that the learning process of our CLSA framework outperforms the ViT-CAT architecture in terms of speed. After approximately 30 epochs, the CLSA framework is adequately trained, while the ViT-CAT requires 50 epochs to reach a similar level of training. Additionally, Fig. 3.26 demonstrates that increasing the cache capacity leads to a higher cache-hit-ratio, with the CLSA framework achieving a superior performance in this aspect.

Finally, we compare the performance of the proposed CLSA architecture with other baselines in terms of the cache-hit ratio. Note that the cache-hit ratio is commonly used in MEC networks to evaluate the effectiveness of the popularity prediction framework. This metric shows the number of requests that are being handled by caching nodes versus the overall number of requests made throughout the network. As shown in Fig. 3.27, there are other baselines in addition to the aforementioned schemes, including Least Recently Used (LRU) [51], Least Frequently Used (LFU) [51], PopCaching [214], and LSTM-C [55]. As depicted

Table 3.12: Comparison with state-of-the-arts.

Model	**Accuracy**	**Model Size**	**Train Time**	**Test Time**	**Memory**
TEDGE (ViT) [5]	93.72%	$4,044,644$	$21.32e-3$	$9.76e-3$	$28e+9$
MTEC [9]	94.13%	$2,882,140$	$17.21e-3$	$6.21e-3$	$27e+9$
ViT-CAT [6]	94.84%	$435,788$	$9.15e-3$	$1.83e-3$	$23e+9$
CoPo [7]	92.99%	$4,993,404$	$6.71e-3$	$2.91e-3$	$19e+9$
DLCC [142]	92.81%	-	-	-	-
Proposed CLSA-Movilens 100k	95.10%	$428,236$	$1.91e-3$	$0.28e-3$	$11e+9$
Proposed CLSA-Movilens 1M	95.42%	$165,036$	$0.98e-3$	$0.11e-3$	$12e+9$

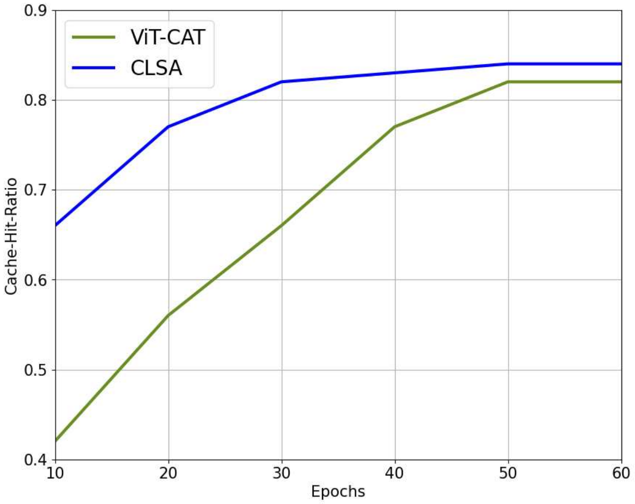


Figure 3.25: A comparison of the cache-hit ratio versus number of epochs.

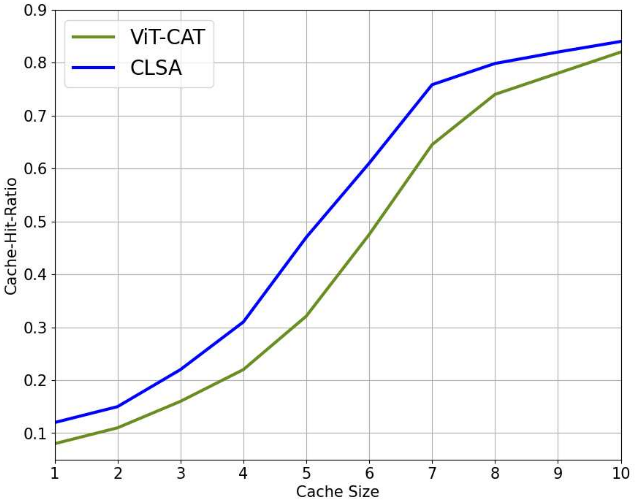


Figure 3.26: A comparison of the cache-hit ratio versus cache capacity.

in Fig. 3.27, the optimal approach [55] is a caching scheme where caching nodes handle all requests throughout the network, which is not feasible in real-world scenarios. Based on the results presented in Fig. 3.27, the proposed CLSA architecture achieves the highest cache-hit ratio when compared to other baselines. It is important to highlight that while the cache-hit ratio of the proposed CLSA framework is comparable to that of the ViT-CAT architecture, it surpasses the ViT-CAT in terms of other metrics including learning speed, model size, running time, and memory usage. As a final note, we would like to elaborate the reason that we used LSTM as the encoder of the proposed CLSA architecture. While Transformers have several advantages, such as parallelization, and their

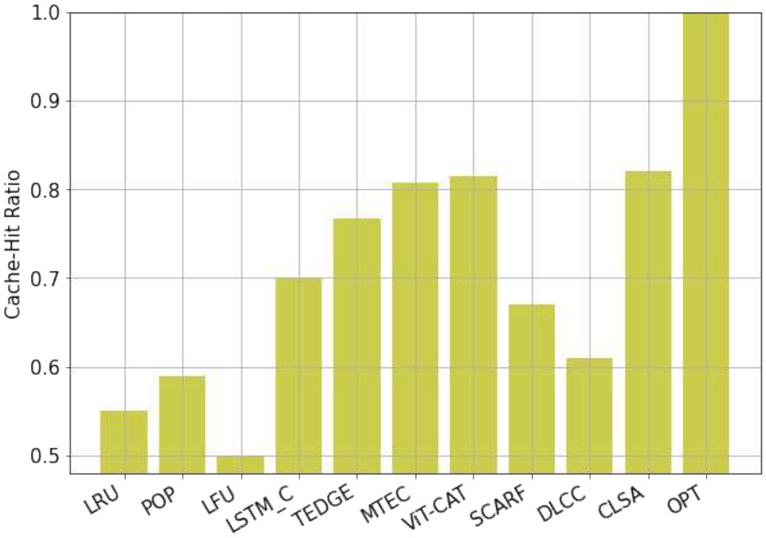


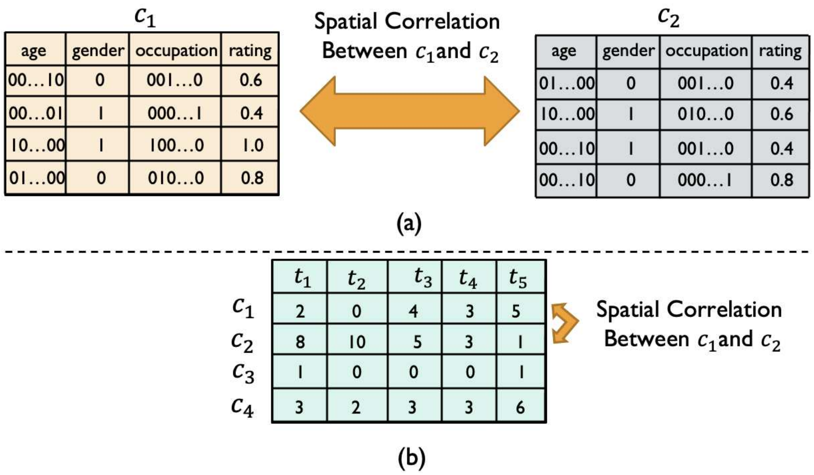
Figure 3.27: Comparison with state-of-the-art based on the cache-hit ratio.

ability to capture long-term dependency, they have the following limitations in comparison to the CLSA architecture:

(i) Transformers, typically, require large datasets for training in order to achieve satisfactory generalization. LSTM models, on the other hand, have shown better performance in scenarios with limited data availability.

(ii) Although Transformers excel in capturing long-term dependencies, a lengthy sequential input is not necessary in the proposed CLSA. In other words, achieving high accuracy is possible by utilizing only 20 timestamps.

(iii) By underscoring the importance of capturing spatial correlation in MEC networks, it is essential to note that, conventional Transformers are incapable of efficiently capturing the spatial correlation among distinct contents in a one-dimensional signal input, where each element corresponds to the number of requests per unit of time for a particular content. ViT can be utilized instead to address this issue (i.e., capturing correlation between N contents) by converting N sets of 1D historical request patterns for each content into 2D images sized $N \times T$. Fig. 3.28 is an example of the input sample of the ViT network, where there are $N = 4$ contents, and the time duration of requests is set to $T = 5$. More precisely, the first row represents the number of requests for content c_1 at time $\{t_1, \ldots, t_5\}$. Since both the request patterns for c_1 and c_2 are captured in a single input sample, it is



age	gender	occupation	rating
00…10	0	001…0	0.6
00…01	1	000…1	0.4
10…00	1	100…0	1.0
01…00	0	010…0	0.8

age	gender	occupation	rating
01…00	0	001…0	0.4
10…00	1	010…0	0.6
00…10	1	001…0	0.4
00…10	0	000…1	0.8

	t_1	t_2	t_3	t_4	t_5
c_1	2	0	4	3	5
c_2	8	10	5	3	1
c_3	1	0	0	0	1
c_4	3	2	3	3	6

Figure 3.28: An illustrative example to represent (a) the spatial correlation between different contents in the proposed CLSA framework, and (b) The ViT network.

possible to extract the spatial correlation between different contents from that single input. This eliminates the requirement of using separate input samples to gather such information. If we, however, were to consider all contents in a single input sample, the input size would become excessively large. For example, with the Movielens Dataset consisting of $1,682$ contents and 20 past requests per content, the input sample size using the proposed CLSA would be 28×20, where 28 denotes the number of user contextual information features. This is 60 times smaller than the input size required by the ViT network, i.e., $1,682 \times 20$, while providing the same classification accuracy. More specifically, spatial correlation in ViT networks is learned using the attention mechanism, for this reason, the historical requests of all contents should be simultaneously fed to the ViT network to attend to different parts of the input sample. This underscores the superiority of the CLSA model in effectively capturing spatial relationships without the need for additional data transformation processes.

3.5 Proposed CoPo Framework

Note that the self-supervised CoPo framework used for pre-training, is very similar to the encoder/decoder part of the CLSA framework, where the shared encoder is taught to map input samples into meaningful latent representations.

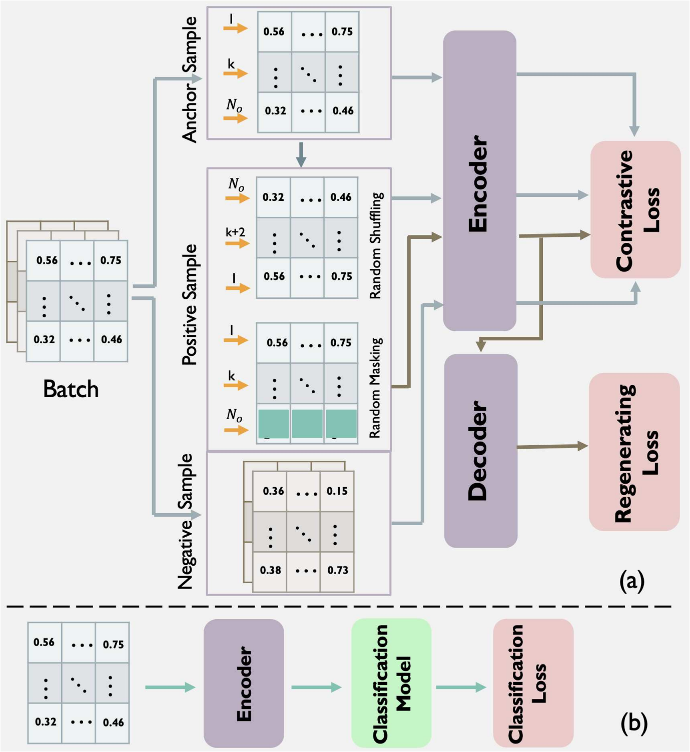


Figure 3.29: The overall perspective of the CoPo architecture.

Following the training of the CL model, a supervised classifier is implemented for fine-tuning. In Fig. 3.29, the encoder that has been trained during the pre-training phase is employed to transform input samples into latent representation. To demonstrate the efficiency of the proposed CoPo framework, a logistic regression model is utilized during the fine-tuning process, where the labels $\{y_k^m\}_{m=1}^M$ are used as the true values.

3.5.1 Simulation Results

A UAV-aided MEC network was studied, comprising four ground-based caching nodes and two aerial ones, and has a total of 943 users and 1682 multimedia items. Based on the typical assumption, the storage capacity of each caching node is 10% of the entire multimedia collection, with all items having identical sizes. The model was trained using a five-fold cross-validation approach, with 80% of the samples being used for training and 20% for testing. The Adam optimizer was employed during training, with betas set at $(0.9, 0.999)$ and weight decay at $1e-7$. To avoid overfitting, l_2 regularization was set at $1e-6$. The

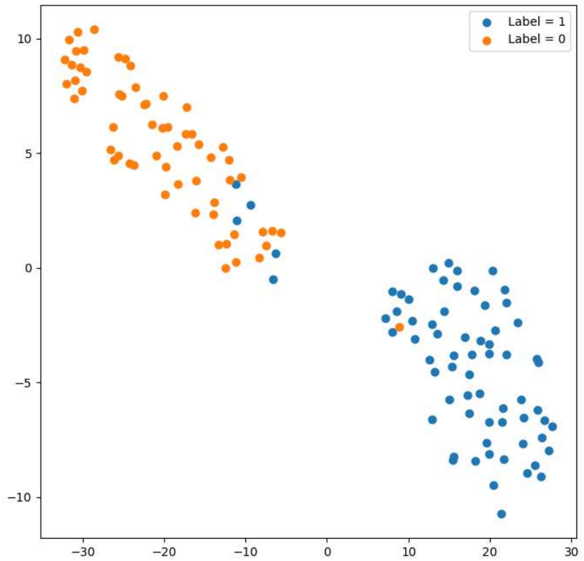


Figure 3.30: A typical embedded space for the latent representation of popular and unpopular content, using The TSNE method.

encoder in this case is implemented using an LSTM network, which utilizes a Rectified Linear Unit (ReLU) and sigmoid activation functions. The output size of this block is indicated by D_{E}. Finally, the decoder performs based on the Time-LSTM2, with an output size of D_{D}, where the activation functions are sigmoid and Tanh. By using a process of trial and error, the best version of the CoPo framework was determined to have the following features: D_{E} and D_{D} were both set to 512, the batch size, the learning rate, and N_o were established at $1e{-}3$, 128, and 20, respectively. Table 3.13 illustrates the accuracy, precision, recall, and f1-score for different 5 folds.

Furthermore, we leverage the TSNE technique [141] to assess the effectiveness of the CL block in producing latent representations that can discriminate between popular and unpopular content. To demonstrate this, we present Fig. 3.30, which illustrates the embedded space of a test set obtained from one of the five-fold cross-validation experiments, which is clearly separable for popular and unpopular contents. Finally, we compare the proposed CoPo framework in terms of classification accuracy with several self-supervised, unsupervised, and supervised learning models. As shown in Table 3.14, the proposed CoPo architecture outperforms other unsupervised baselines, i.e., Adaptive Genetic Neural

Network (AGNN) [143] and Artificial Neural Network (ANN) with modified K-Means [143], and Self-Supervised Contrastive Learning using Random Feature Corruption (SCARF) [144]. Moreover, we compare the proposed CoPo framework with several supervised learning models, such as Vision Transformer [5], Multiple-model Transformer-based Edge Caching (MTEC) [9], Vision Transformers with Cross Attention (ViT-CAT) [6], and Deep Learning-based Content Caching (DLCC) [142] frameworks. Table 3.14 demonstrates that the classification accuracy is comparable to that of supervised learning models while eliminating the need for manual labeling of datasets, which saves time.

3.6 Conclusion

In this Chapter, we targeted the problem of autonomous connection scheduling using RL-based architectures in Section 3.1 by introducing the CQN-CS framework. Given an optimal connection scheduling framework, we focused on data-driven popularity prediction models adapted with uncoded content placement in MEC networks. We first proposed the attention-based TEDGE caching framework in Section 3.2, which is based on ViT network. Despite the fact that the TEDGE framework achieves a high level of classification accuracy, it was intricate and demands a significant number of parameters in order to achieve such accuracy. In Section 3.3, we presented a low-complex and parallel ViT-CAT fusion architecture to predict the Top-K popular contents in MEC networks. One significant limitation of supervised popularity prediction models is the requirement for manual labeling of contents as popular or unpopular by investigating users' past behavior, which can be a time-intensive task. Sections 3.4 and 3.29 proposed self-supervised learning algorithms called Contrastive to predict the dynamic content popularity in a MEC network in a self-supervised manner.

Table 3.13: Accuracy, precision, recall, and f1-score for two classes (i.e., popular (class 1) and unpopular (class 0)) using 5 fold cross-validation.

	Class	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**
Accuracy	-	95.19 ± 1.05	92.17 ± 0.71	91.62 ± 0.23	93.54 ± 0.45	92.44 ± 0.23	92.99 ± 1.44
Precision	**0**	97.96 ± 1.10	94.76 ± 1.41	92.68 ± 0.58	92.72 ± 0.95	98.43 ± 0.60	95.31 ± 2.68
	1	92.74 ± 1.76	89.87 ± 0.88	90.61 ± 0.45	94.39 ± 0.06	87.77 ± 0.46	91.08 ± 2.49
Recall	**0**	92.30 ± 2.00	89.28 ± 1.05	90.38 ± 0.54	94.50 ± 0.01	86.26 ± 0.63	90.54 ± 3.01
	1	98.07 ± 1.05	95.05 ± 1.41	92.85 ± 0.63	92.58 ± 1.05	98.62 ± 0.54	95.43 ± 2.74
F1-score	**0**	95.04 ± 1.28	91.93 ± 0.83	91.51 ± 0.27	93.60 ± 0.48	91.94 ± 0.30	92.81 ± 1.51
	1	95.33 ± 1.16	92.38 ± 0.82	91.72 ± 0.28	93.47 ± 0.56	92.88 ± 0.25	93.16 ± 1.40

Table 3.14: Comparison with state-of-the-art based on the classification accuracy.

Model	Self-Supervised/Unsupervised Learning				Supervised Learning			
	CoPo	AGNN [143]	ANN + Modified K-Means [143]	SCARF [144]	ViT	MTEC	ViT-CAT	DLCC [142]
Accuracy	92.99%	81%	74%	87.17%	93.72%	94.13%	94.84%	92.81%

Chapter 4

Popularity Prediction and Coded/Uncoded Content Placement

Using the CQN-CS architecture, introduced in Chapter 3, we improved the network's coverage and provided a highly reliable and low-latency transmission for outdoor users. Nevertheless, there is currently no effective infrastructure that can also benefit indoor users. Referred to as the Cluster-centric and Coded UAV-aided Femtocaching (CCUF) framework, the network's coverage in both indoor and outdoor environments increases by considering a two-phase clustering framework for FAPs' formation and UAVs' deployment. Due to the UAV's signal attenuation in indoor environments, we consider two different indoor and outdoor caching service scenarios (see Fig. 4.1) for the proposed CCUF framework. More precisely, the indoor area is covered by FAPs, equipped with extra storage and supported by Coordinated Multi-Point (CoMP) technology. The outdoor area, however, is supported by coupled UAVs and FAPs depending on the movement speed of ground users. To access a large number of content during the movement of ground users, a two-phase clustering approach is considered: (i) The whole network (both indoor and outdoor areas) is partitioned into sub-networks called inter-clusters, which is defined for content placement in FAPs. We show that based on this strategy, the ground users can acquire more segments during their movements, and; (ii) For UAVs formation, the outdoor environment is partitioned into intra-clusters via K-means clustering scheme, each covered by a UAV.

Another goal of the CCUF framework is to increase content diversity by using a cluster-centric cellular network, where multimedia contents are classified into three categories, including popular, mediocre, and non-popular contents. While the popular contents are stored completely, distinct segments of mediocre ones are determined according to the proposed framework to be stored in the storage of neighboring FAPs. We also determine the best number of coded/uncoded contents in each caching node to increase the cache-hit-ratio, SINR, and cache diversity while decreasing users' access delay and cache redundancy for different popularity profiles.

Finally, we introduce a novel attention-based popularity prediction framework adopted with the coded/uncoded content placement. More precisely, with the assumption that users' preferences remain unchanged over a short horizon, the Top-K popular contents are identified as the output of the learning model. Most existing data-driven popularity prediction models, however, are not suitable for the coded/uncoded content placement frameworks. On the one hand, in coded/uncoded content placement, in addition to classifying contents into two groups, i.e., popular and non-popular, the probability of content request is required to identify which content should be stored partially/completely, where this information is not provided by existing data-driven popularity prediction models. On the other hand, the assumption that users' preferences remain unchanged over a short horizon only works for content with a smooth request pattern. To tackle these challenges, we develop a Multiple-model (hybrid) Transformer-based Edge Caching (MTEC) framework with higher generalization ability, suitable for various types of content with different time-varying behavior, that can be adapted with coded/uncoded content placement frameworks.

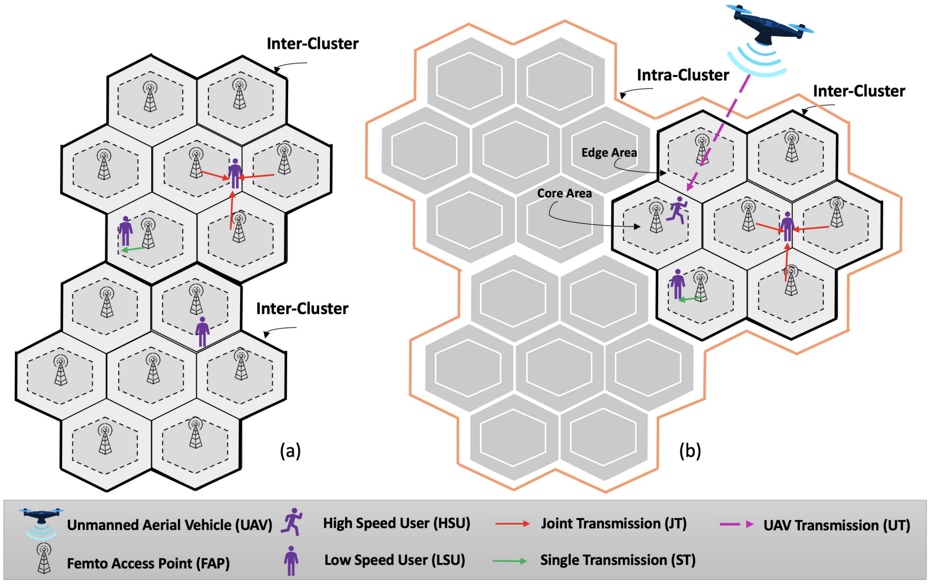


Figure 4.1: A typical structure of the proposed UAV-aided cellular network in (a) the indoor, and (b) the outdoor environments.

4.1 Coded/Uncoded Content Placement

In this Section, we consider an integrated UAV-aided and cluster-centric cellular network to serve ground users positioned in both indoor and outdoor environments. Our first objective is to increase the content diversity that can be accessed via caching nodes. The second goal is to introduce different transmission schemes for indoor/outdoor users to improve the achievable QoS in terms of the users' access delay and decrease the energy consumption of UAVs. Referred to as the Cluster-centric and Coded UAV-aided Femtocaching (CCUF) framework, the network's coverage in both indoor and outdoor environments increases by considering a two-phase clustering approach for FAPs' formation and UAVs' deployment. In summary, this work makes the following key contributions:

- Due to the UAV's signal attenuation in indoor environments, we consider two different indoor and outdoor caching service scenarios for the proposed CCUF framework. More precisely, the indoor area is covered by FAPs, equipped with extra storage and supported by CoMP technology. The outdoor area, however, is supported by coupled UAVs and FAPs depending on the movement speed of ground users.
- To access a large number of content during the movement of ground users, a

two-phase clustering approach is considered: (*i*) The whole network (both indoor and outdoor areas) is partitioned into sub-networks called inter-clusters, which is defined for content placement in FAPs. We show that based on this strategy, the ground users can acquire more segments during their movements, and; (*ii*) For UAVs formation, the outdoor environment is partitioned into intra-clusters via a K-means clustering algorithm, each covered by a UAV.

- To the best of our knowledge, despite all the research conducted in this field, there is no placement strategy to determine how distinct segments of popular content should be distributed in different caching nodes. Towards this goal, we consider a cluster-centric cellular network, where multimedia contents are classified into three categories, including popular, mediocre, and non-popular contents. While the popular contents are stored completely, distinct segments of mediocre ones are determined according to the proposed framework to be stored in the storage of neighboring FAPs. We also determine the best number of coded/uncoded contents in each caching node to increase the cache-hit-ratio, SINR, and cache diversity while decreasing users' access delay and cache redundancy for different content popularity profiles.

4.1.1 System Model and Problem Description

We consider a UAV-aided cellular network in a residential area that supports both indoor and outdoor environments (see Fig. 4.1). There exist N_f number of FAPs, denoted by f_i, for $(1 \leq i \leq N_f)$, each with the cache size of C_f and transmission range of R_f. All FAPs are independently and randomly distributed in the environment following Poisson Point Processes (PPPs) [45]. There are also N_u number of UAVs, denoted by u_k, for $(1 \leq k \leq N_u)$, with equal transmission range of R_u, and a main server that has access to the whole content and can manage all caching nodes. There are N_g number of ground users, denoted by GU_j, for $(1 \leq j \leq N_g)$, that move through the network with different velocities. Term $\upsilon_j(t)$ denotes the speed of the ground user GU_j at time slot t. When GU_j requests content c_l from a library of $\mathcal{C} = \{c_1, \ldots, c_{N_c}\}$, in which $N_c = |\mathcal{C}|$ is the cardinality of multimedia data in the network, this request should be handled by one of the nearest FAPs or UAVs having some segments of c_l. In this work, FAP f_i, for $(1 \leq i \leq N_f)$, and UAV u_k, for $(1 \leq k \leq N_u)$, operate in an open access mode, i.e., they can serve any ground user GU_j, for $(1 \leq j \leq N_g)$, located in their transmission range. To completely download a requested content, a finite time T is required. In the proposed CCUF framework, each content c_l is fragmented into N_s encoded segments, denoted by c_{ls}, for $(1 \leq s \leq N_s)$. Without loss of generality, it is assumed that the time T is discretized into N_s time slots with time interval δ_t, i.e., $T = N_s\delta_t$, where δ_t is large enough for downloading one segment c_{ls}. A summary of the notations used hereinafter is provided in Table 4.1.

As it can be seen from Fig. 4.1, we consider two different indoor and outdoor caching service scenarios for the proposed CCUF framework to improve the network's coverage. As will be described in Subsection 4.1.3, the requests of indoor users are handled through FAPs, while outdoor users are supported by coupled UAVs and FAPs depending on their movement speed. In this regard, we define two clustering approaches, called inter-clusters and intra-clusters, which are used for content placement in FAPs and UAVs' deployment, as discussed below:

Inter-Clusters: As shown in Fig. 4.1, $N_b < N_f$ number of neighboring FAPs in both indoor and outdoor environments, form a cluster, referred to as the inter-cluster. As it is stated in [45, 145], the main focus of cluster-centric content placement is to place contents in the storage of FAPs, where all FAPs belonging to an inter-cluster are used as an entity (despite conventional femtocaching schemes where each FAP acts as single storage). Therefore, our goal is to determine how different segments of popular files should be distributed in the cache of FAPs belonging to an inter-cluster to increase the content diversity. We construct the inter-clusters based on the following two rules: (i) As will be described shortly in Subsection 4.1.2, all FAPs in the same inter-cluster save *different* segments of mediocre contents, and; (ii) The cached contents of different inter-clusters are the *same*. In addition, all FAPs use the CoMP transmission approach (supporting ST and JT schemes) to mitigate the inter-cell interference in edge areas and manage ground users' requests.

Table 4.1: List of Notations.

Notation	**Description**	**Notation**	**Description**
N_g	Number of ground users	N_0	Noise power
N_f	Number of FAPs	$\mathcal{L}_{i,j}(t)$	Path loss between f_i and GU_j
N_b	Number of FAPs in each inter-cluster	$\mathcal{L}_{k,j}^{(LoS)}(t)$	LoS path loss from u_k to GU_j
N_u	Number of UAVs	χ_σ	Shadowing effect
N_c	Number of contents	η	Path loss exponent
N_s	Number of encoded segments	$\eta^{(LoS)}$	Path loss exponent of LoS
R_f	FAP's transmission range	$d_{i,j}(t)$	Distance between f_i and GU_j
R_u	UAV's transmission range	$d_{k,j}(t)$	Distance between u_k and GU_j
C_f	FAP's cache size	f_c	Carrier frequency
C_u	Cache size of UAV u_k	c	Light speed
T	Download time duration for a content	$p_{k,j}^{(LoS)}(t)$	Probability of LoS link u_k and GU_j
δ_t	Download time duration for a segment	$v_j(t)$	Speed of GU_j at time slot t
h_k	Altitude of UAV u_k	$p_u^{(h)}(t)$	Probability of cache-hit at time slot t
p_l	Probability of requesting content c_l	$p_u^{(m)}(t)$	Probability of cache-miss at time slot t
γ	Popularity Skewness	$\mathcal{D}_u(t)$	User's access delay through UAVs
r	Rank of the file c_l	$\mathcal{D}_f(t)$	User's access delay through FAPs
α	Storage portion for popular content	L_c	File size of c_l
$\mathcal{S}_{i,j}$	SINR from f_i to GU_j	$R_{k,j}$	Transmission data rate
P_i	Transmitted signal power of FAP f_i	P_k	Transmission power of UAV u_k
$I_{\boldsymbol{f}_{-i}}(t)$	Interference power of FAP-ground users	$\mathbf{y}$	Indicator vector for popular content
β	Cache capacity	$\mathbf{z}$	Indicator vector for mediocre content
ψ	Requests served by UAVs	ζ	Normalized Users' velocity

Intra-Clusters: Since the transmission range of FAPs is significantly less than that of a UAV (see Fig. 4.1(b)), the outdoor area is also divided into several intra-clusters (each intra-cluster is covered by a UAV) based on an unsupervised learning algorithm. In what follows, we present the content popularity profile and transmission schemes utilized to develop the proposed CCUF framework.

4.1.2 Content Popularity Profile

To account for user's behavior pattern in multimedia services, the popularity of video contents is determined based on the Zipf distribution [44, 146], where the probability of requesting the l^{th} file, denoted by p_l, is calculated as

$$p_l = \frac{l^{-\gamma}}{\sum\limits_{r=1}^{N_c} r^{-\gamma}}, \tag{67}$$

where γ and r represent the skewness of the file popularity, and the rank of the file c_l, respectively. For notational convenience, we assume that $P[n] \equiv P(n\delta_t)$ denotes the probability of accessing a new segment in time slot n, with $n \in \{1, \ldots, N_s\}$. Without loss of generality and to be practical, we investigate the probability distribution of a real multimedia data set, i.e., the YouTube videos trending statistics, following Zipf distribution, i.e., a small part of the contents are requested with a high probability. The majority of contents are not popular, and some contents, are requested moderately. Consequently, in the proposed CCUF framework, we classify multimedia contents into three categories, i.e., popular, mediocre, and non-popular [45]. To improve content diversity, the storage capacity of FAPs, denoted by C_f, is divided into two spaces, where α portion of the storage is allocated to store complete popular contents, i.e., $1 \leq l \leq \lfloor \alpha C_f \rfloor$, where $l = 1$ indicates the most popular content. Additionally, $(1 - \alpha)$ portion of the cache is assigned to store different parts of the mediocre contents, where $\lfloor \alpha C_f \rfloor + 1 \leq l \leq N_s(C_f - \lfloor \alpha C_f \rfloor)$. The best value of α is obtained experimentally. The proposed model for identifying different segments to be cached in neighboring FAPs will be discussed later on in Subsection 4.1.4.

4.1.3 Transmission Scheme

In this Subsection, we describe both the connection scheduling (serving by FAPs or UAVs) and the transmission scheme depending on the presence of the ground user in indoor or outdoor environments.

Indoor Environment

The transmitted signal by UAVs, propagating in residential areas, becomes weaker due to the penetration loss and shadow fading effects. It is, therefore, assumed that ground users positioned in indoor areas are only supported by FAPs. In the CoMP-integrated and cluster-centric cellular network and as it can be seen from Fig. 4.1, there are two regions in each inter-cluster, named cell-edge and cell-core, which are determined based on the long-term averaged SINR values [57] to illustrate the quality of a wireless link. In such a case that the ground user GU_j is positioned in the vicinity of the FAP f_i, the SINR from f_i to GU_j, denoted by $\mathcal{S}_{i,j}$, is obtained as follows

$$\mathcal{S}_{i,j}(t) = \frac{P_i|\tilde{\mathcal{H}}_{i,j}(t)|^2}{I_{\boldsymbol{f}_{-i}}(t) + N_0}, \tag{68}$$

where P_i denotes the transmitted signal power of FAP f_i, and $I_{\boldsymbol{f}_{-i}}(t)$ represents the interference power from other FAP-ground users, except for the corresponding f_i link. Term N_0 represents the noise power related to the additive white Gaussian random variable. Moreover, the path loss and fading channel effects between FAP f_i and ground user GU_j at time slot t is denoted by $\tilde{\mathcal{H}}_{i,j}(t) = \dfrac{h_{i,j}(t)}{\sqrt{\mathcal{L}_{i,j}(t)}}$. In this case, $h_{i,j}(t)$ denotes a complex zero-mean Gaussian random variable with unit standard deviation and $\mathcal{L}_{i,j}(t)$ represents the path loss between FAP f_i and ground user GU_j at time slot t, obtained as follows

$$\mathcal{L}_{i,j}(t) = \mathcal{L}_0 + 10\eta \log\big(d_{i,j}(t)\big) + \chi_\sigma, \tag{69}$$

where η is the path loss exponent. Term χ_σ indicates the shadowing effect, which is a zero-mean Gaussian-distributed random variable with standard deviation σ. Additionally, $d_{i,j}(t)$ represents the Euclidean distance between FAP f_i and ground user GU_j at time slot t. Furthermore, $\mathcal{L}_0 = 20\log\left(\frac{4\pi f_c d_0}{c}\right)$ is the path loss related to the reference distance d_0 where f_c and $c = 3 \times 10^8$ denote the carrier frequency and the light speed, respectively. Accordingly, the ground user GU_j is marked as the cell-core user connected to FAP f_i, if $\overline{\mathcal{S}}_{i,j}(t) > \mathcal{S}_{th}$; otherwise, GU_j is marked as the cell-edge user, where $\mathcal{S}_{th}$ is the SINR threshold.

The transmission scheme in the proposed CoMP-integrated and cluster-centric cellular network is determined based on two metrics; (i) The popularity of the requested content, described in Subsection 4.1.2, and; (ii) The link quality of the ground user in the cell, i.e., cell-core or cell-edge. The following two different transmission schemes are utilized for the development of the proposed CCUF framework:

- **Single Transmission (ST):** In this case, the requested file c_l, for $(1 \leq l \leq \lfloor \alpha C_f \rfloor)$, is a popular content, and the ground user GU_j is marked as the cell-core of FAP f_i, i.e., $\overline{\mathcal{S}}_{i,j}(t) > \mathcal{S}_{th}$. It means that the content is completely cached into the storage of FAP f_i and the high-quality link can be established between FAP f_i and ground user GU_j. Consequently, this request is served only by the corresponding FAP f_i. Moreover, if the requested content belongs to the mediocre category, i.e., $\lfloor \alpha C_f \rfloor + 1 \leq l \leq N_s(C_f - \lfloor \alpha C_f \rfloor)$, this request is served according to the ST scheme regardless of the user's link quality since each FAP has a different segment of the mediocre content.
- **Joint Transmission (JT):** In this transmission scheme, the requested file c_l is a popular content, i.e., $1 \leq l \leq \lfloor \alpha C_f \rfloor$. Consequently, all FAPs have the same complete file. The ground user GU_j, however, is marked as the cell-edge of FAP f_i, i.e., $\overline{\mathcal{S}}_{i,j}(t) \leq \mathcal{S}_{th}$. Therefore, the link quality between FAP f_i and the ground user GU_j is not good enough. In order to improve the reliability of content delivery, the corresponding content will be jointly transmitted by several FAPs in its inter-cluster. As it can be seen from Fig. 4.1, neighboring FAPs in an inter-cluster collaboratively serve cell-edge ground users based on the JT scheme, which is shown by the red color. Using such a specific indoor transmission scheme, we take the advantage of the CoMP technology to further decrease the users' access delay by

simultaneously serving mediocre content via multiple caching nodes.

Outdoor Environment

As it can be seen from Fig. 4.1, outdoor areas are covered by both UAVs and FAPs. Therefore, outdoor users are classified based on their velocity into the following two categories:

- **Low Speed Users (LSUs):** If the speed of ground user GU_j, denoted by $\upsilon_j(t)$, is less than a predefined threshold υ_{th}, this user is managed by inter-clusters (FAPs). Therefore, the transmission scheme of LSUs is completely the same as the indoor users, described in Subsection 4.1.3.
- **High Speed Users (HSUs):** In this case, the speed of ground user GU_j is equal or more than υ_{th}. Therefore, this request should be served by a UAV covering the corresponding intra-cluster.

This completes our discussion on the content popularity profile and transmission schemes. Next, we develop the CCUF framework.

4.1.4 The CCUF Framework

In conventional femtocaching schemes, it is a common assumption that all caching nodes store the same most popular contents [131, 147, 148]. This assumption is acceptable in static femtocaching models, in which users are stationary or move with a low velocity. With the focus on a dynamic femtocaching network, in which users can move based on the random walk model, storing distinct content in neighboring FAPs leads to increasing the number of requests served by caching nodes [149]. Despite recent researches on cluster-centric cellular networks, there is no framework to determine how different segments should be stored to increase content diversity. Toward this goal, we propose the CCUF framework, which is an efficient content placement strategy for the network model introduced in Subsection 4.1.1. The proposed CCUF framework is implemented based on the steps presented in the following.

Content Caching for FAPs and UAVs

Identifying the best multimedia content to be stored in the storage of caching nodes leads to a reduction in the users' latency. It is commonly assumed [44,131] that the total users' access delay is determined according to the availability of the required content in the nearby caching nodes. Based on this assumption, in scenarios where the requested content can be served by caching nodes, the cache-hit occurs and the ground user will experience no delay; otherwise, the request is served by the main server resulting in a cache-miss. Inspired from [150], we relax the above assumption and express the actual users' access delay as a function of the content popularity profile and link's quality. In this regard, we propose two optimization models for content placement in both FAPs and UAVs to minimize the users' latency, which are similar in nature. Toward this goal, we first describe the delay that ground users experience when served by FAPs and UAVs.

UAVs' Content Placement: Serving requests via UAVs leads to establishing air-to-ground links from UAVs to ground users. Due to the obstacles in outdoor environments, the transmitted signal from UAVs is attenuated. To be practical, we consider both LoS and Non-LoS (NLoS) path losses from UAV u_k to ground user GU_j at time slot t as follows [170]

$$\begin{aligned}\mathcal{L}_{k,j}^{(LoS)}(t) =&\ \mathcal{L}_0 + 10\eta^{(LoS)} \log(d_{k,j}(t)) + \chi_\sigma^{(LoS)}, &(70)\\ \mathcal{L}_{k,j}^{(NLoS)}(t) =&\ \mathcal{L}_0 + 10\eta^{(NLoS)} \log(d_{k,j}(t)) + \chi_\sigma^{(NLoS)}, &(71)\end{aligned}$$

where $\mathcal{L}_0 = 20\log\left(\frac{4\pi f_c d_0}{c}\right)$ denotes the reference path loss in distance d_0, and $d_{k,j}(t)$ is the Euclidean distance between UAV u_k and the ground user GU_j at time slot t. In addition, $\eta^{(LoS)}$, $\eta^{(NLoS)}$, $\chi_\sigma^{(LoS)}$ and $\chi_\sigma^{(NLoS)}$ indicate the LoS and NLoS path loss exponents and the corresponding shadowing effects, respectively. Consequently, the average path loss, denoted by $\overline{\mathcal{L}}_{k,j}(t)$, is given by

$$\overline{\mathcal{L}}_{k,j}(t) = p_{k,j}^{(LoS)}(t)\overline{\mathcal{L}}_{k,j}^{(LoS)}(t) + (1 - p_{k,j}^{(LoS)}(t))\overline{\mathcal{L}}_{k,j}^{(NLoS)}(t), \qquad (72)$$

where $p_{k,j}^{(LoS)}(t)$ is the probability of establishing LoS link between UAV u_k and ground user GU_j at time slot t, obtained as [169]

$$p_{k,j}^{(LoS)}(t) = (1 + \vartheta \exp(-\zeta[\phi_{k,j}(t) - \vartheta]))^{-1}, \tag{73}$$

where ϑ and ζ are constant parameters, depending on the rural and urban areas. Moreover, $\phi_{k,j}(t) = \sin^{-1}\left(\frac{h_k}{d_{k,j}(t)}\right)$ is the elevation angle between UAV u_k and the ground user GU_j, and h_k is the UAV's altitude. Without loss of generality, altitude h_k is assumed to be a fixed value over the hovering time. If the requested content cannot be found in the storage of UAVs, additional ground-to-air connection is required to provide UAVs with the requested content through the main server. Similarly, the average path loss of the main server-to-UAV u_k link is calculated as

$$\overline{\mathcal{L}}_{m,k}(t) = p_{m,k}^{(LoS)}(t)\mathcal{L}_{m,k}^{(LoS)}(t) + (1 - p_{m,k}^{(LoS)}(t))\mathcal{L}_{m,k}^{(NLoS)}(t), \tag{74}$$

where $\mathcal{L}_{m,k}^{(LoS)}(t) = d_{m,k}^{-\varpi}(t)$ and $\mathcal{L}_{m,k}^{(NLoS)}(t) = \psi\mathcal{L}_{m,k}^{(LoS)}(t)$, in which $d_{m,k}(t)$ denotes the distance between the main server and UAV u_k. Furthermore, ϖ and ψ denote the LoS and NLoS path loss exponents, respectively [170].

As stated previously, another parameter that has a great impact on the users' access delay is the presence of the requested content in the caching node, depending on the content popularity profile. Therefore, the cache-hit and the cache-miss probability through serving by UAV u_k at time slot t, denoted by $p_u^{(h)}(t)$ and $p_u^{(m)}(t)$, respectively, are expressed as

$$\begin{aligned} p_u^{(h)}(t) &= \sum_{l \in C_u} p_l(t) \leq 1, & (75) \\ p_u^{(m)}(t) &= 1 - p_u^{(h)}(t), & (76) \end{aligned}$$

where C_u denotes the cache size of UAV u_k, which is assumed to be the same for all UAVs. Consequently, the users' access delay through UAVs is expressed as

$$\mathcal{D}_u(t) = p_u^{(h)}(t)\mathcal{D}_u^{(h)}(t) + p_u^{(m)}(t)\mathcal{D}_u^{(m)}(t), \tag{77}$$

where $\mathcal{D}_u^{(h)}(t)$ and $\mathcal{D}_u^{(m)}(t)$ represent the cache-hit and the cache-miss delays, respectively, calculated as follows [169]

$$\mathcal{D}_u^{(h)}(t) = \frac{L_c}{R_{k,j}} = L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, \boldsymbol{u}_{-k}) + N_0}\right), \tag{78}$$

$$\begin{aligned} \mathcal{D}_u^{(m)}(t) = & \underbrace{L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{m,k}(t)/10}}{I_k(t, \boldsymbol{u}_{-k}) + N_0}\right)}_{\triangleq \mathbf{L_{MU}}} + \\ & \underbrace{L_c \log^{-1}\left(1 + \frac{P_k 10^{\overline{\mathcal{L}}_{k,j}(t)/10}}{I_k(t, \boldsymbol{u}_{-k}) + N_0}\right)}_{\triangleq \mathbf{L_{UG}}}, \end{aligned} \tag{79}$$

where L_c and $R_{k,j}$ represent the file size of c_l and the transmission data rate from UAV u_k to GU_j. Furthermore, P_k and $I_k(t, \boldsymbol{u}_{-k})$ denote the transmission power of UAV u_k and the interference power caused by other UAV-user links for the transmission link between u_k and GU_j, respectively. Note that when the cache-miss happens, the content should be first provided for the UAV by the main server. Therefore, L_{MU} and L_{UG} in Eq. (79) represent the users' access delay related to the main server-UAV and UAV-ground user links, respectively.

Given users' access delay through UAVs, the goal is to place contents in the storage of UAVs to minimize the users' access delay in Eq. (77). Due to the large coverage area of UAVs, it is not feasible for ground users to move through areas supported by different UAVs frequently. Therefore, we assume that contents (either popular or mediocre ones) are cached completely in the storage of UAVs. With the aim of minimizing users' access delay, the cached contents are selected as the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathrm{x}_l} \quad & \sum_{l=1}^{N_c} \Big(\sum_{j=1}^{N_g} p_l^{(j)}(t) \mathcal{D}_u^{(j)}(t) \Big) \mathrm{x}_l \\ \text{s.t.} \quad & \mathbf{C1.} \;\; \mathrm{x}_l \in \{0, 1\}, \\ & \mathbf{C2.} \;\; \sum_{l=1}^{N_c} (1 - \mathrm{x}_l) \leq C_u, \end{aligned} \tag{80}$$

where $p_l^{(j)}(t)$ denotes the probability of requesting content c_l by the ground user GU_j at time slot t, which is obtained according to the request history of ground user GU_j [131]. Furthermore, $\mathcal{D}_u^{(j)}(t)$ is the delay that the ground user GU_j may experience, which is calculated based on Eq. (77). In the constraint **C1**, x_l is an indicator variable, which is equal to 0 when content c_l exists in the cache of UAV u_k. Moreover, the constraint **C2** represents that the total contents cached in the storage of u_k should not exceed its storage capacity of u_k.

FAPs' Content Placement: Serving requests by FAPs leads to a ground-to-ground connection type between FAPs and ground users. Similarly, the users' access delay through FAP connections is calculated as

$$\mathcal{D}_f(t) = p_k^{(h)}(t)\mathcal{D}_f^{(h)}(t) + p_k^{(m)}(t)\mathcal{D}_f^{(m)}(t), \tag{81}$$

where $\mathcal{D}_f^{(h)}(t)$, as the cache-hit delay, is expressed as

$$\mathcal{D}_f^{(h)}(t) = \frac{L_c}{R_{i,j}} = L_c \log^{-1}\left(1 + \frac{P_i|\tilde{\mathcal{H}}_{i,j}(t)|^2}{I_{\boldsymbol{f}_{-i}}(t) + N_0}.\right) \tag{82}$$

In this case, coded contents to be stored in the storage of FAPs are determined according to the solution of the following optimization problem:

$$\begin{aligned} \mathcal{F}(\mathbf{y},\mathbf{z}) \;=\; & \min_{y_l,z_l} \;\; \sum_{l=1}^{N_c}\Big(\sum_{j=1}^{N_g} p_l^{(j)}(t)\mathcal{D}_f^{(j)}(t)\Big)y_l \\ & + \sum_{l=1}^{N_c}\Big(\sum_{j=1}^{N_g} p_l^{(j)}(t)\mathcal{D}_f^{(j)}(t)\Big)z_l, \\ \text{s.t.} \quad & \mathbf{C1.}\;\; y_l, z_l \in \{0,1\}, \\ & \mathbf{C2.}\;\; \sum_{l=1}^{N_c}(1-y_l) \le \lfloor \alpha C_f \rfloor, \\ & \mathbf{C3.}\;\; \sum_{l=1}^{N_c}(1-z_l) \le N_s(C_f - \lfloor \alpha C_f \rfloor), \end{aligned} \tag{83}$$

where $\mathcal{F}(\mathbf{y},\mathbf{z})$ is the cost function associated with users' access delay, experienced by serving the request through FAPs. By assuming that $N_p = \lfloor \alpha C_f \rfloor$ and $N_a = N_s(C_f - \lfloor \alpha C_f \rfloor)$ are the cardinality of popular and mediocre contents, respectively, $\mathbf{y} = [y_1, \ldots, y_{N_p}]^T$ is an indicator vector for popular contents, where y_l would be 0 if l^{th} content is stored in the cache of FAPs, otherwise it equals

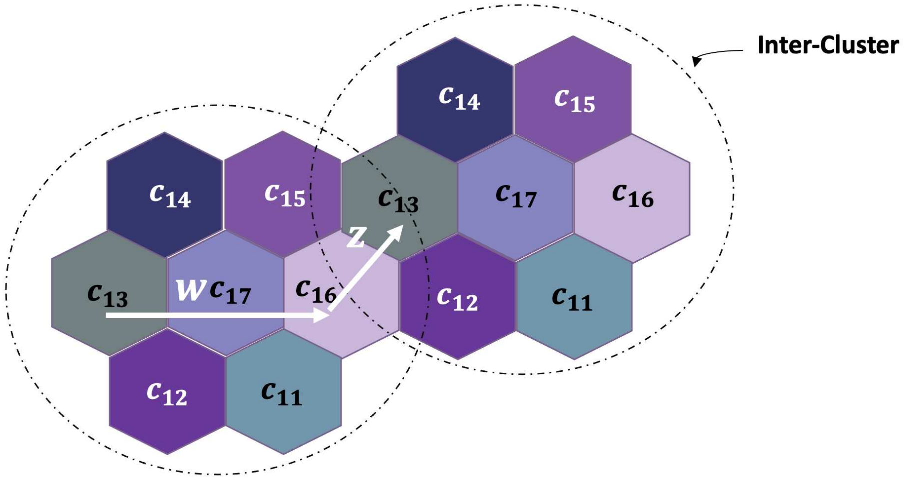


Figure 4.2: A typical hexagonal cellular network, where seven FAPs form an inter-cluster.

to 1. Similarly, $\mathbf{z} = [z_1, \ldots, z_{N_a}]^T$ is an indicator variable for mediocre contents. According to the optimization problem, different from popular contents that are completely stored, just one segment of mediocre contents are cached. Similarly, y_l and z_l in constraint **C1** illustrate the availability of content c_l in the cache of FAP f_i. Finally, constraints **C2** and **C3** indicate the portion of cache allocated to popular and mediocre contents, respectively. Due to the large size of video contents and the complexity of the content placement, it is essential to update the storage of caching nodes in the off-peak period [151]. Therefore, we use an adaptive time window for cache updating, introduced in our previous work [131], to maintain a trade-off between the on-time popularity recognition of contents and the network's traffic.

Content Placement in Multiple Inter-Clusters

After identifying popular and mediocre contents, we need to determine how to store different segments of mediocre contents within (i) An inter-cluster, and; (ii) Multiple inter-clusters.

Single Inter-Cluster: The main idea behind the coded placement scheme in our proposed CCUF framework comes from the frequency reusing technique in cellular networks [211]. The distance between two cells with the same spectrum bandwidth is determined in such a way that the resource availability increases and the inter-cell interference decreases [213]. With the same argument in [211, 213], the same mediocre contents are stored in different inter-clusters, while different FAPs belonging to an inter-cluster store different segments of the mediocre contents. Without loss of generality, we first consider a simple hexagonal cellular network including N_b FAPs as one inter-cluster (Fig. 4.2). Given the vector $\mathbf{z} = [z_1, \dots, z_{N_a}]^T$ that determines the mediocre contents, in this phase, we need to indicate which segment of the mediocre content c_l, denoted by c_{ls} for $(1 \leq l \leq N_a)$ and $(1 \leq s \leq N_s)$, should be cached in FAP f_i for $(1 \leq i \leq N_b)$. In this regard, we form an $(N_a \times N_s)$ indicator matrix of FAP f_i, denoted by $\boldsymbol{Z}^{(f_i)}$, where the l^{th} row of $\boldsymbol{Z}^{(f_i)}$, denoted by $\boldsymbol{z}_l^{(f_i)} = [0, 0, \dots, 1]_{(1 \times N_s)}$ indicates segments of file c_l stored in the cache of FAP f_i. Note that $\boldsymbol{z}_l^{(f_i)}$ is a zeros vector with only one non-zero element, where $z_{ls}^{(f_i)} = 1$ means that s^{th} segment of file c_l is stored in the cache of FAP f_i. To store different segments of mediocre contents within an inter-cluster, the cached contents of FAP f_j for $(1 \leq j \leq N_b, j \neq i)$ in the inter-cluster is determined as follows

$$\boldsymbol{z}_l^{(f_i)} \boldsymbol{z}_l^{(f_j)^T} = 0, \quad i = 1, \dots, N_b, j = 1, \dots, N_b, i \neq j. \tag{84}$$

Multiple Inter-Clusters: After allocating mediocre contents to FAPs inside an inter-cluster, the same content as FAP f_i is stored in FAP f_k in the neighboring inter-cluster, where k is given by

$$\boldsymbol{Z}^{(f_k)} = \boldsymbol{Z}^{(f_i)} \quad \text{if} \quad k = w^2 + wz + z^2, \tag{85}$$

where w and z represent the number of FAPs required to reach another FAP storing similar contents, in two different directions [213] (see Fig. 4.2). More precisely, first, it is required to move w cell along any direction from FAP f_i, then turn 60 degrees counter-clockwise and move z cells to reach FAP f_k. For example, in Fig. 4.2, $N_s = 7$, $w = 2$, and $z = 1$ for starting in a FAP including c_{13} and reaching a similar FAP in a neighboring inter-cluster.

Remark 1: In a practical scenario, the coverage area of FAPs is influenced by path loss and shadowing models (it is not a hexagonal shape). Location $p = (x, y)$ is placed within the transmission area of FAP f_i, if the strength of the received signal at point p, denoted by $RSSI_p$, is higher than the threshold value $RSSI_{th}$, where $RSSI_p$ is calculated as

$$RSSI_p(dB) = RSSI(d_0) + 10\eta \log_{10}(\frac{d}{d_0}) + X_\sigma, \tag{86}$$

with d and d_0 denoting the distance between FAP and point p in the boundary of transmission area of FAP, and the reference distance is set to 1 [m], respectively. Moreover, η represents the path loss exponent, which is 10 [dB] or 20 [dB], and X_σ is a zero-mean Gaussian with standard deviation σ that represents the effect of multi-path fading in the CCUF scheme [154].

Success Probability in the Proposed CCUF Framework

To quantify the benefits of the proposed CCUF strategy, we define success probability, which is defined as the probability of finding a new segment by user GU_j at time slot t under the following two scenarios: (i) Uncoded cluster-centric, and; (ii) Coded cluster-centric UAV-aided femtocaching network, denoted by p_{uc}, and p_{cc}, respectively. Concerning the nature of mobile networks, ground users move and leave their current positions. In this work, it is assumed that low-speed ground users can obtain one segment in each contact, i.e., $T = N_s\delta_t$ is required to completely download content c_l. First, we consider a simple mobility, where ground users are positioned in the transmission area of a new FAP in each time slot t. Eventually, in $T = N_s\delta_t$, the whole content c_l will be downloaded. Then, we generalize the mobility of ground users to the random walk model, where ground users can return to their previous place.

Simple Movement Scenario: Regarding the uncoded cluster-centric UAV aided femtocaching framework, content c_l consisting of c_{ls}, for $(1 \leq s \leq N_s)$ segments, is stored completely in all FAPs. Consequently, the probability of downloading $n = N_s$ segments of content c_l in $T = N_s\delta_t$ depends on the probability of requesting file c_l, denoted by p_l. Since the storage capacity of each FAP is equal to C_f, the success probability, denoted by p_{uc}, is obtained as follows

$$p_{uc}[n = N_s, t = T] = \sum_{l=1}^{C_f} p_l. \tag{87}$$

Algorithm 2 Proposed CCUF Strategy

```
1: Initialization: Set $\alpha$, $\lambda$, $N_s$, and $C_f$.
2: Input: $p_l^{(j)}(t)$.
3: Output: $\mathrm{x}_l$, $\mathrm{y}_l$, and $\mathrm{z}_l$.
4: Content Placement Phase:
5: for $u_k, k = 1, \ldots, N_u$, do
       $\min_{\mathrm{x}_l} \quad \sum_{l=1}^{N_c} \Big( \sum_{j=1}^{N_g} p_l^{(j)}(t)\mathcal{D}_u^{(j)}(t) \Big) \mathrm{x}_l$
6:         s.t.   C1. and C2. in Eq. (80).
7: end for
8: for $f_i, i = 1, \ldots, N_f$, do
       $\min_{y_l, z_l} \quad \sum_{l=1}^{N_c} \Big( \sum_{j=1}^{N_g} p_l^{(j)}(t)\mathcal{D}_f^{(j)}(t) \Big) y_l +$
       $\sum_{l=1}^{N_c} \Big( \sum_{j=1}^{N_g} p_l^{(j)}(t)\mathcal{D}_f^{(j)}(t) \Big) z_l,$
9:         s.t.   C1.-C3. in Eq. (83).
10: end for
11: $\boldsymbol{z}_l^{(f_i)} \boldsymbol{z}_l^{(f_j)^T} = 0, \quad i = 1, \ldots, N_s, j = 1, \ldots, N_s, i \neq j,$
12: $\boldsymbol{Z}^{(f_k)} = \boldsymbol{Z}^{(f_i)} \quad \text{if} \quad k = w^2 + wz + z^2,$
13: Transmission Phase:
14: for $GU_j, j = 1, \ldots, N_g$, do
15:     if $GU_j$ is in indoor environment then
16:         if $GU_j$ is an edge-user and requests
17:         popular content then
18:             The request should be handled according to the
19:             JT scheme.
20:         else
21:             The request should be handled according to the
22:             ST scheme.
23:         end if
24:     else
25:         if $v_j(t) \geq v_{\mathrm{th}}$ then
26:
27:             The request is served by UAV $u_k$.
28:         else
29:             Similar to lines 16 to 23.
30:         end if
31:     end if
32: end for
```

On the other hand, the success probability of the CCUF framework is obtained as

$$p_{cc}[n = N_s, t = T] = \sum_{l=1}^{N_p} p_l + \sum_{l=N_p+1}^{N_a+N_p} p_l. \tag{88}$$

To illustrate the growth rate of the success probability in the coded one, we rewrite p_{uc} in Eq. (87) as follows

$$p_{uc}[n = N_s, t = T] = \sum_{l=1}^{N_p} p_l + \sum_{l=N_p+1}^{C_f} p_l. \tag{89}$$

As it can be seen from Eqs. (88), and (89), the first term related to the popular content is the same. The second term, however, illustrates that the number of distinct contents that can be served through FAPs within an inter-cluster in the coded cluster-centric network is $\varkappa$ times greater than the uncoded one, where $\varkappa$ is given by

$$\varkappa = \frac{\lfloor \alpha C_f \rfloor + N_s(C_f - \lfloor \alpha C_f \rfloor)}{C_f}. \tag{90}$$

Accordingly, due to the allocation of different segments in the coded cluster-centric network, more segments of the desired contents are accessible during the users' movement in the simple movement scenario. Therefore, more requests can be served in comparison to the uncoded cluster-centric UAV-aided femtocaching networks.

Generalizing to Random Walk Scenario: In contrary to the simple movement scenario discussed above, the following two situations are possible where the ground user GU_j cannot find a new segment during its movement: (i) Returning back to the previous coverage area of FAPs; and, (ii) Positioning in the transmission area of a FAP, which stores the same segment of the content that the ground user has already downloaded. Consequently, the success probability of the coded cluster-centric will not be the same as the previous scenario. If the requested content is the popular one, regardless of the link's quality of the ground user within an inter-cluster, the ground user can download one segment of the required content with the probability of $\sum_{l=1}^{N_p} p_l$ at each contact. While this part of the success probability is constant, the success probability of downloading a new segment of a mediocre content in each contact depends on the current and previous locations of the ground user. Therefore, we first determine the success probability of achieving a new segment of a mediocre content, denoted by $p_{ns}(n = n_0, t = n_0\delta_t)$, for $(1 \leq n_0 \leq N_s)$. Then, we calculate the success probability of a coded cluster-centric network based on the random walk movement.

As it can be seen from Fig. 4.2, regardless of the location of GU_j, this user can download one segment successfully in the first contact (i.e., $n_0 = 1$). Therefore, we have $p_{ns}(n = 1, t = \delta_t) = 1$. Similarly, when $n_0 = 2$, the ground user GU_j can download a new segment without considering the location of the ground user. Therefore, the probability of downloading two segments after two contacts is $p_{ns}[n = 2, t = 2\delta_t] = 1$. More precisely, in the second contact, the ground user can be positioned in the cell of $(N_s - 1)$ number of FAPs, where the probability of being in the cell of FAP f_i is $p(f = f_i) = \dfrac{1}{(N_s - 1)}$. Therefore, we have

$$\begin{aligned} p_{ns}[n = 2, t = 2\delta_t] \ &= \sum_{i=1}^{N_s-1} p_{ns}[n = 2, t = 2\delta_t | f = f_i] p(f = f_i) \\ (N_s - 1) \times 1 \times \frac{1}{N_s - 1} &+ 0 \times \frac{1}{N_s - 1} = 1. \end{aligned} \tag{91}$$

Accordingly, the probability of finding a new segment in the third contact is obtained as follows

$$\begin{aligned} p_{ns}[n = 3, t = 3\delta_t] &= \sum_{i=1}^{N_s-1} p_{ns}[n = 3, t = 3\delta_t | f = f_i] p(f = f_i) \\ = (N_s - 2)\frac{1}{N_s - 1} &\times 1 + \frac{1}{N_s - 1} \times 0 = \frac{N_s - 2}{N_s - 1}, \end{aligned} \tag{92}$$

where $(N_s - 2)$ FAPs have different segments, whereas if GU_j returns to the FAP at $t = \delta_t$, the ground user can find a similar segment. Similarly, it can be proved that the probability of finding a new segment in $n > 2$ is given by

$$\begin{aligned} p_{ns}[n = n_0, t = n_0\delta_t] &= \sum_{i=1}^{N_s-1} p_{ns}[n = n_0, t = n_0\delta_t | f = f_i] \\ \times p(f = f_i) &= \frac{(N_s - 2)^{n_0-2}}{(N_s - 1)^{n_0-2}}, \qquad \text{for } n_0 > 2. \end{aligned} \tag{93}$$

Taking into account the unequal likelihood of finding new segments of mediocre contents in different contacts, we recalculate p_{cc} as follows

$$\begin{aligned} p_{cc}[n = N_s, t = T] &= \sum_{l=1}^{N_p} p_l \\ &+ \sum_{n=1}^{N_s} \frac{(N_s - 2)^{n-2}}{(N_s - 1)^{n-2}} \Big(\sum_{l=N_p+1}^{N_s(C_f - \lfloor \alpha C_f \rfloor)} p_l \Big). \end{aligned} \tag{94}$$

Remark 2: Regarding the users' mobility pattern, the proposed CCUF framework is not limited to the random mobility pattern, and is also effective for vehicles moving at a constant speed. Intuitively speaking, we focus on human mobility patterns, which can be more complex than that of the constant velocity model. In other words, people can walk around in different directions, changing directions, and/or return back to a previously visited location. To account for such scenarios, we consider the random mobility pattern that allows movement in different directions. According to the success probability, expressed in Eqs. (88) and (94), the proposed CCUF framework is efficient for both constant velocity and random movement scenarios.

Remark 3: In such a case that the location of GU_j in two consecutive time slots is the same, it means that GU_j is a fixed user. Therefore, the following two scenarios can happen depending on the popularity of the requested content: (i) Similar to the mobile users' case, if the requested content c_l is popular, the whole segments of file c_l are sent by neighboring FAP to GU_j, and; (ii) If c_l is a mediocre content, each FAP within the inter-cluster transmits one segment of file c_l, which is known as the Parallel Transmission (PT) [45].

3-D Deployment of UAVs in Intra-clusters

To increase the resource availability for ground users, the outdoor environment is partitioned based on an unsupervised learning algorithm, where each partition is covered by a UAV. Considering a Gaussian mixture distribution for ground users, we have a dense population of ground users in some areas. The main goal is to deploy UAVs in such a way that ground users can experience high QoS communications even in a dense area. Towards this goal, we obtain the 3-D UAV placement based on the following two phases. First, the altitude of UAVs is determined according to Reference [155] to find the optimum deployment height. Then, the K-means clustering algorithm, which has been already employed in our proposed CCUF, is applied to determine the optimal 2-D position. Considering the relationship between the UAV's altitude, their transmission range, and the maximum allowable path-loss, the optimal height of the UAV $h^{(opt)}$, is obtained as follows [155]

$$h^{(opt)} = R_u^{(opt)} \tan \phi_{opt}, \tag{95}$$

where ϕ_{opt} represents the optimal elevation angle maximizing the coverage radius, which is a constant depending on the environment [156]. In [157], the wireless network model is extended from a single UAV [155] to the multi-UAV scenario, stating that the transmission coverage should be a fraction of the optimal value to avoid/reduce the interference between UAVs. Accordingly, the optimal coverage range of UAVs $R_{u,m}^{(opt)}$ in a multi-UAV scenario is expressed as [157]

$$R_{u,m}^{(opt)} = \varrho R_u^{(opt)}, \tag{96}$$

where $0 < \varrho \leq 1$ is a constant parameter [157] depending on the number of UAVs in the network. The value of ϱ decreases by increasing the number of UAVs, therefore, the optimal value of UAVs' height in multi-UAV scenarios is calculated as

$$h_{u,m}^{(opt)} = \frac{R_{u,m}^{(opt)}}{\tan(\Theta_B/2)}, \tag{97}$$

where Θ_B denotes the antenna beamwidth. After finding the optimal height of the UAV, the goal is to partition N_g ground users into K intra-clusters, where the sum of Euclidean distances between the ground user GU_j, for $(1 \leq j \leq N_g^k)$, and UAV u_k is minimized. In this case, N_g^k is the cardinality of ground users positioned in the intra-cluster related to the UAV u_k. Therefore, the UAVs' deployment is obtained according to the following optimization problem

$$\min_{\boldsymbol{l}_k(t)} \quad \sum_{k=1}^{N_u} \sum_{j=1}^{N_g^k} ||\boldsymbol{l}_j(t), \boldsymbol{l}_k(t)||, \tag{98}$$

where $\boldsymbol{l}_k(t)$ denotes the location of the UAV u_k at time slot t, defined as the mean of the coordinates of all ground users inside the corresponding intra-cluster as follows

$$\boldsymbol{l}_k(t) = \frac{\sum_{j=1}^{N_g^k} \boldsymbol{l}_j(t)}{N_g^k}, \qquad k = 1, \ldots, N_u. \tag{99}$$

To solve the above optimization problem, we utilize the K-Means clustering algorithm [158], which is known as an efficient unsupervised learning framework. In the first step, a set of points, denoted by $\mathcal{P} = \{P_1, \ldots, P_{N_u}\}$, is generated, where P_k for $(1 \leq k \leq N_u)$ should be within the pre-specified environment. Then, the set of ground users in the vicinity of P_k is determined as follows

$$u_j \in N_g^k \quad \text{if} \;\; ||\boldsymbol{l}_j(t), P_k|| < ||\boldsymbol{l}_j(t), P_r||, \;\; \forall k \neq r. \tag{100}$$

Given the set of ground users belonging to each intra-cluster, UAVs' locations are determined according to Eq. (99). In the second step, by moving ground users from one intra-cluster to another, the Euclidean distances between ground users and UAVs are calculated to update the location of UAVs according to Eq. (98). The K-Means algorithm is terminated when there is no change in the ground users belonging to an intra-cluster over several iterations. It should be noted that it is common [159, 170] to consider height optimization and horizontal placement separately. The focus of our ongoing research is on joint 3-D optimization, which is a fruitful direction for future research. This completes our discussion on development of the CCUF scheme. The pseudo-code of the proposed CCUF framework is summarized in **Algorithm 2**.

4.1.5 Simulation Results

To demonstrate the advantage of the proposed CCUF framework, we consider a UAV-aided cellular network with $R = 5000$ [m], covered by the main server. There are $N_f = 175$ FAPs, and $N_u = 10$ UAVs, where each inter-cluster compromises of $N_s = 7$ FAPs. Considering the fact that hovering posture consumes less energy [160], within the context of UAV-aided cellular networks, it is a common assumption [161, 162] that UAVs remain hovering at their locations while serving a request for data delivery. As quadcopters can easily change their positions with the least amount of bending angle [163, 164], and can hover in their locations while serving a request, we consider quadcopters as the UAVs in the proposed CCUF framework. Without loss of generality and for simplicity, we consider a static clustering scheme where there are a fixed number of FAPs in each inter-cluster to determine how different segments of popular files should be distributed in an inter-cluster to increase the content diversity. Therefore, the number of FAPs in each inter-cluster is considered to be the same as the number of segments N_s. We consider the path loss threshold $\mathcal{L}_{th} = 100$ [dB] [165] to guarantee a high level of QoS for ground users. Following Reference [165], the optimal elevation angle ϕ_{opt} in a dense urban area is 54.62°. Consequently, the optimal transmission range R_u and optimal height of the UAV for a single-UAV scenario are computed as 430 [m] and 640 [m], respectively [165]. In a multi-UAV scenario with $N_u = 10$ and $\Theta_B = 100°$ [157], the optimal coverage range $R_{u,m}^{(opt)} = 0.261 R_u$ and the optimal altitude are 112 [m] and 94 [m], respectively [157]. According to the restrictions of the aviation regulations of different countries, UAVs can fly horizontally up-to a predefined maximum height $h^{(\max)}$. For example, in United States $h^{(\max)} = 122$ [m], and in Australia $h^{(\max)} = 120$ [m] [166]. Therefore, the feasible optimal altitude for UAV u_k is equal to $h_k = \min\{h^{(\max)}, h_{u,m}^{(\text{opt})}\}$ [167]. The general simulation parameters are summarized in Table 5.9. As it is proved in [44, 131] that the optimum content placement is an NP-hard problem, we use *fmincon* optimization toolbox, implemented in MATLAB (R2020a), to solve Eqs. (80) and (83).

Fig. 4.3 depicts an integrated heterogeneous network, where yellow and red areas determine indoor and outdoor environments, respectively. Fig. 4.3 also shows the deployment of UAVs in the intra-clusters within the network, which is generated by partitioning ground users according to the K-means clustering algorithm. As a result of the Gaussian mixture distribution for clients, we have a dense population in some areas, which can be changed over time by the movement of ground users. Therefore, the location of $N_u = 10$ UAVs and the formation of intra-clusters in this work is varying, depending on the user density distribution. For the comparison purpose and in order to find the best value of

Table 4.2: List of Parameters.

Notation	**Value**	**Notation**	**Value**
N_g	500	$\eta^{(LoS)}$, $\eta^{(NLoS)}$	2.5, 3
N_f	180	P_k	30 [dBm]
N_u	10	ϖ, ψ	2, 20
N_s	7	L_c	37.5 [MB]
N_c	40724	τ_p	$0-5$ [ms]
R_f	30 [m]	$\chi_\sigma^{(LoS)}$, $\chi_\sigma^{(NLoS)}$	1.6, 23
$P_T(t)$, $P_R(t)$	0.5 , 0.25 [W]	N_0	-94 [dBm]

α, three types of caching strategies are considered:

- *Uncoded UAV-aided Femtocaching (UUF):* This scheme is derived by modifying the Fairness Scheduling algorithm with an Adaptive Time Window (FS-ATW) scheme [131], where the proposed content placement strategy in [131] is used for both UAVs and FAPs. The popular contents in the UUF model are stored completely into FAPs and UAVs without any coding and clustering schemes. Therefore, it is equivalent to our proposed CCUF framework, where the value of α, which indicates the percentage of contents stored completely, would be one (i.e., $\alpha = 1$).
- *Proposed Cluster-centric and Coded UAV-aided Femtocaching (CCUF):* In this case, the uncoded popular and the coded mediocre contents are stored in the caching nodes, where $0 < \alpha < 1$. According to the simulation results, the best value of α is obtained.
- *The Conventional Cluster-centric and Coded UAV-aided Femtocaching (Conventional CCUF):* This scheme is an upgraded version of the FemtoCaching scheme in [44], integrated with the CoMP technology. In this framework, regardless of the content popularity profile, all contents are stored partially. For simplicity, this scheme is shown by $\alpha = 0$ in simulation results.

These three strategies are evaluated over the cache-hit-ratio, cache diversity, cache redundancy, SINR, and users' access delay to determine the best value of α. Moreover, to illustrate the effect of considering a UAV-aided femtocaching framework in an integrated network, we compare the users' access delay and energy consumption of UAVs, by serving users in both indoor and outdoor areas.

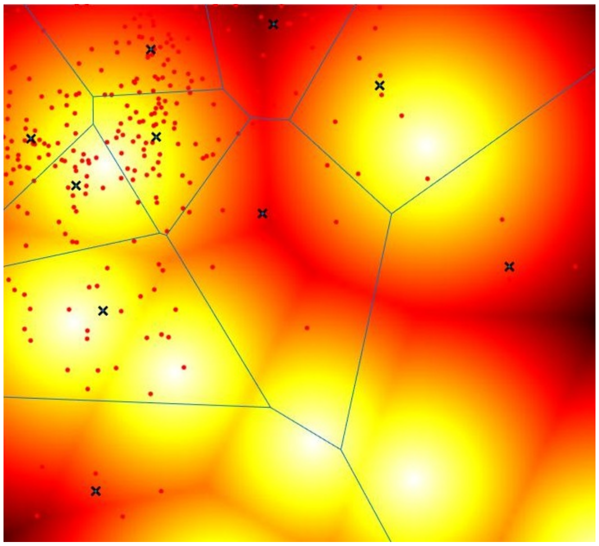


Figure 4.3: Deployment of UAVs in intra-clusters within an integrated network, where “yellow” and “red” colors indicate indoor and outdoor environments, respectively.

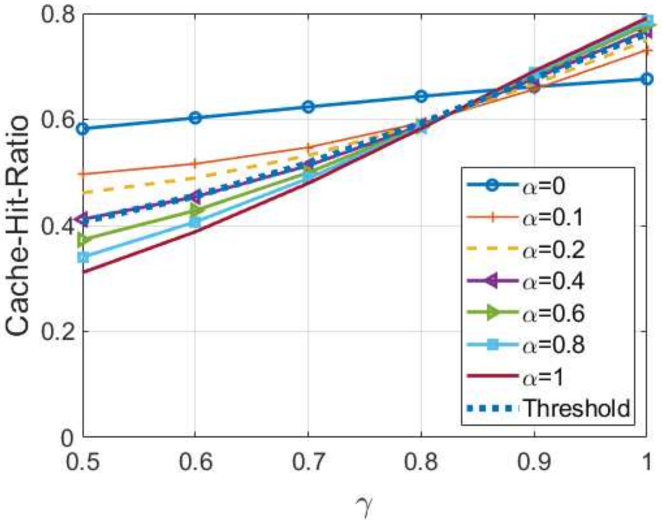


Figure 4.4: The cache-hit-ratio versus the popularity parameter γ for different values of α.

Cache-Hit-Ratio: This metric illustrates the number of requests served by caching nodes versus the total number of requests made across the network. The high value of cache-hit-ratio shows the superiority of the framework. Since we assume that ground users can download one segment in each contact, we evaluate the cache-hit-ratio in terms of the number of fragmented contents served by caching nodes. Fig. 4.4 compares the cache-hit-ratio of the UUF ($\alpha = 1$), the proposed CCUF ($0 < \alpha < 1$), and conventional CCUF ($\alpha = 0$) frameworks versus the value of γ. As previously mentioned, parameter γ shows the skewness of

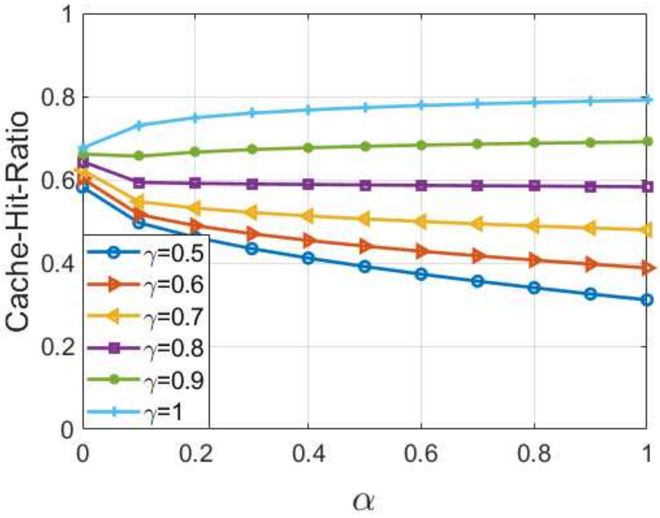


Figure 4.5: The cache-hit-ratio versus the α percentage of contents that are stored completely.

the content popularity, where $\gamma \in [0, 1]$. Note that the large value of γ indicates that a small number of contents has a high popularity, where a small value of γ illustrates an almost uniform popularity distribution for the majority of contents. As it can be seen from Fig. 4.4, depending on the popularity distribution of contents, γ, the conventional CCUF framework results in a higher cache-hit-ratio. The most important reason is that given a constant cache capacity, the coded content placement of the conventional CCUF strategy leads to a remarkable surge in the content diversity. In contrast, for a high value of γ, where a small number of contents is widely requested, the UUF and the proposed CCUF frameworks have better results compared to the conventional CCUF. By considering the fact that the common value of γ is about $0.5 \leq \gamma \leq 0.6$ (e.g., see [2,44]), we define CHR_{th} as the threshold cache-hit-ratio, which is the average of cache-hit-ratio of different values of α for a specific γ. As it can be seen from Fig. 4.4, the proposed CCUF framework with $0 < \alpha \leq 0.4$ and the UUF scheme outperform other schemes from the aspect of cache-hit-ratio.

Fig. 4.5 shows the cache-hit-ratio versus different values of α when the popularity parameter γ changes in the range of 0.5 to 1. Accordingly, for $0.5 \leq \gamma \leq 0.6$, by increasing the value of α, the cache-hit-ratio decreases drastically. In the following, we also investigate the impact of α on the users' access delay to determine the best value of α.

Users' Access Delay: Users' access delay depends on three parameters, i.e., the

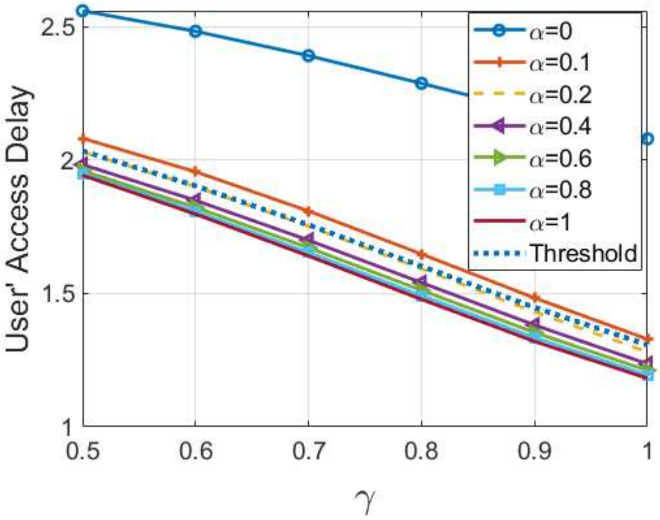


Figure 4.6: The users' access delay in the indoor environment versus different value of γ.

availability of the content in caching nodes, the distance between the ground user and the corresponding caching node, and the channel quality, known as the SINR. Figs. 4.6 and 4.7 compare the users' access delay of the aforementioned frameworks, which is obtained according to Eq. (81). By utilizing the CoMP technology in the proposed CCUF, serving edge-users according to the JT scheme has a great impact on the SINR, where users' access delay decrease by increasing the SINR. As can be seen from Table 4.5, the SINR of edge-users improves by increasing the value of α. Note that JT scheme can be performed if the same contents are stored in the neighboring FAPs. Therefore, by increasing the value of α, the users' access delay will decrease. With the same argument, we define $\mathcal{D}_{th}$, which is the average of users' access delay of different values of α for a specific γ, shown in Fig. 4.6. Therefore, the best value of α would be $\alpha \geq 0.2$. Consequently, the cache-hit-ratio and users' access delay of the proposed CCUF framework would be efficient if $\alpha \in [0.2, 0.4]$.

Cache Diversity: This metric illustrates the diversity of contents in an inter-cluster, which is defined as the number of distinct segments of contents, expressed as follows

$$\mathcal{CD} = \frac{N_a}{N_s C_f} = 1 - \frac{\lfloor \alpha C_f \rfloor}{C_f}. \tag{101}$$

As stated previously, we have $N_a = N_s(C_f - \lfloor \alpha C_f \rfloor)$. As it can be seen from

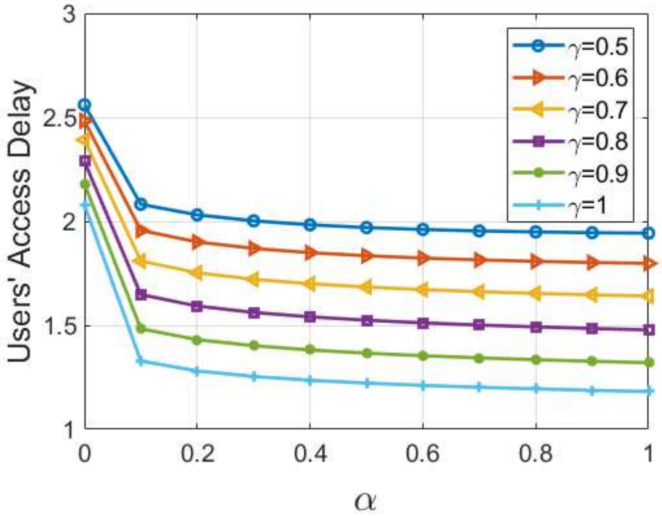


Figure 4.7: The users' access delay in the indoor environment versus different values of α.

Table 4.3: The SINR experienced by edge-users for different values of α and γ.

	$\gamma = 0.5$	$\gamma = 0.6$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 1$
$\alpha = 0$	16.37	16.37	16.37	16.37	16.37	16.37
$\alpha = 0.1$	17.55	18.12	18.89	19.84	20.88	21.88
$\alpha = 0.2$	18.01	18.65	19.46	20.40	21.38	22.30
$\alpha = 0.4$	18.62	19.30	20.11	21.00	21.90	22.70
$\alpha = 0.6$	19.06	19.75	20.53	21.37	22.20	22.93
$\alpha = 0.8$	19.42	20.09	20.85	21.64	22.42	23.08
$\alpha = 1$	19.72	20.38	21.11	21.86	22.58	23.20

Fig. 4.8, the value of $\mathcal{CD}$ would be one, if $\alpha = 0$, which means that all cached contents are different. The cache diversity, however, linearly decreases by increasing the value of α, and reaches the lowest value zero, when all contents are cached completely (i.e., $\alpha = 1$).

Cache Redundancy: This metric indicates the number of similar contents that ground users meet during their random movements. As it can be seen from Fig. 4.8, the cache redundancy increases by storing the entire contents. By considering the coded content placement, even in the proposed CCUF framework, ground users that move randomly through the network, can meet a similar coded contents during their movements.

Maximum Required Cache Capacity: Given a specific number of contents through the network, denoted by N_c, the storage capacity of caching nodes is determined

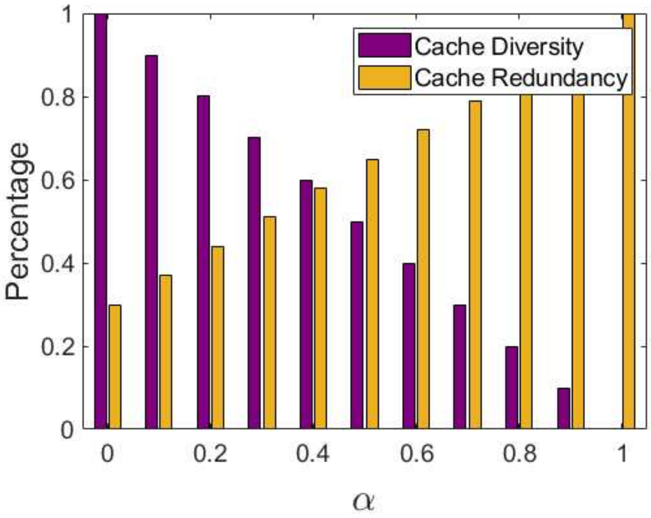


Figure 4.8: The percentage of the cache diversity and the cache redundancy versus different values of α.

by $C_f = \beta N_c$. In this case, parameter β indicates the percentage of contents that can be stored in caching nodes. In the coded content placement, since only one segment of the contents is cached, it is fairly likely that the total number of possible segments that can be cached exceeds the total number of contents. Therefore, the maximum required cache capacity, denoted by β_{max}, for different values of α is obtained as

$$\beta_{max} \leq \frac{N_c}{N_s N_c - (N_s - 1)\alpha N_c} = \frac{1}{\alpha(1 - N_s) + N_s}, \tag{102}$$

where the remainder of the storage would be occupied by redundant contents if $\beta > \beta_{max}$. As it can be seen from Fig. 4.9, the maximum cache capacity β_{max} increases by the value of α. Consequently, in smaller values of α, we need a smaller cache capacity to have the maximum cache diversity.

Users' Access Delay through UAVs and UAVs' Energy Consumption: We evaluate the users' access delay and the energy consumption of UAVs in Figs. 4.10 and 4.11 when the ground user is located in both indoor and outdoor environments. As can be seen from Fig. 4.11, serving requests through UAVs, especially in such a case that ground users are located in the indoor environment, leads to consuming the energy of UAVs, calculated as follows [26]

$$E_{u_k}^{(LoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(LoS)}(t)(\tau_f - \tau_p), \tag{103}$$

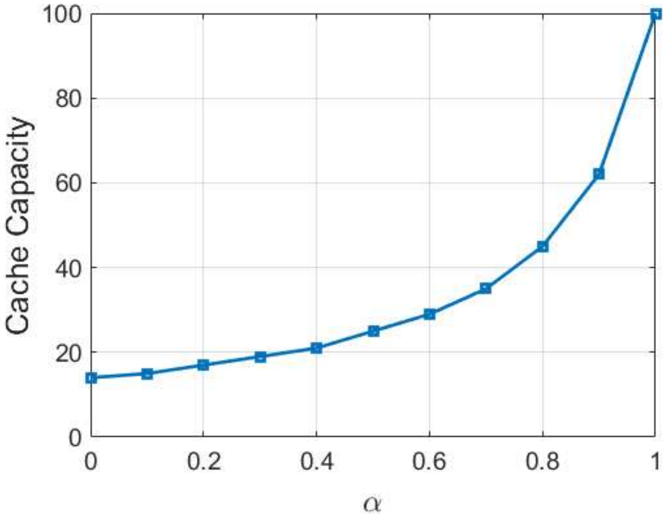


Figure 4.9: The maximum cache capacity, required to achieve the maximum cache diversity, versus different values of α.

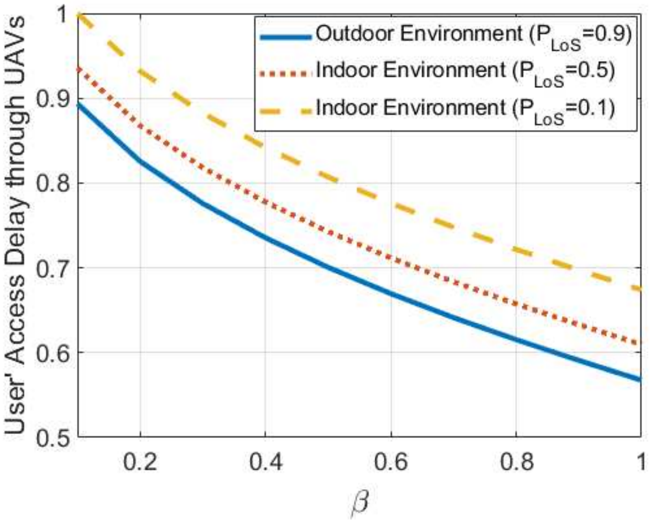


Figure 4.10: Normalized users' access delay experienced through UAVs in both indoor and outdoor versus different values of β.

$$E_{u_k}^{(NLoS)}(t) = L_c P_T(t)\tau_p + L_c P_R(t)\tau_p + P_j^{(NLoS)}(t)(\tau_f - \tau_p), \tag{104}$$

where $P_T(t)$ and $P_R(t)$ represent the power consumed for transmission and reception powers of 1 [Mb] file, respectively. Moreover, $P_j(t)$, τ_f, and τ_p denote the received power at ground user GU_j, and the flyby and the pause times of UAV u_k, respectively. On the other hand, it can be shown from Fig. 4.10 that the indoor users being served through UAVs, experience higher delay in comparison with outdoor users. Consequently, it can be seen that serving indoor

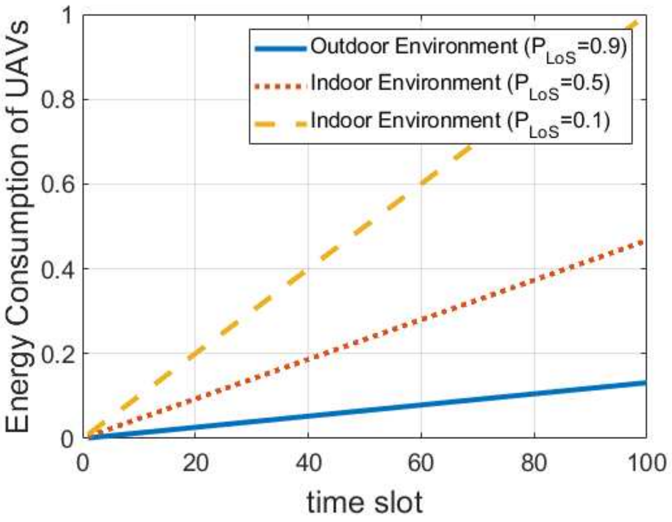


Figure 4.11: Normalized energy consumption of UAVs in both indoor and outdoor environments in different time slots.

users by UAVs could not be efficient from the aspect of user's access delay and energy consumption of UAVs. For this reason, ground users in indoor areas are served by FAPs in inter-clusters.

Finally, Figs. 4.12 and 4.13 illustrate the advantage of serving outdoor users with both FAPs and UAVs in the CCUF framework. Fig. 4.12 compares the average normalized UAVs' energy consumption in different scenarios, where ψ illustrates the ratio of requests served by UAVs to the whole outdoor users' requests. For instance, $\psi = 1$ means all requests in an outdoor environment are managed by UAVs regardless of the user's velocity, while in $\psi = 0.7$, it is assumed that 70% of outdoor users are HSUs, who are supported by UAVs and 30% of users are LSUs, managed by FAPs. As it can be seen from Fig. 4.12, serving LSUs' requests by FAPs leads to a reduction in UAVs' energy consumption. Considering the fact that UAVs are limited energy caching nodes, expanding UAVs' lifetime is of paramount importance. Moreover, Fig. 4.13 illustrates the effect of users' velocity on the FAP's handover probability, where $\zeta = \dfrac{\upsilon}{\upsilon_{\text{th}}}$. For comparison purposes, two scenarios are defined, where *FAP Connection* is associated with a case that all outdoor users regardless of their velocities are supported by FAPs, while in *UAV Connection*, LSUs and HSUs are supported by FAPs and UAVs, respectively. In this case, handover is triggered if the ground user leaves the current FAP's coverage before completely downloading one segment. As it

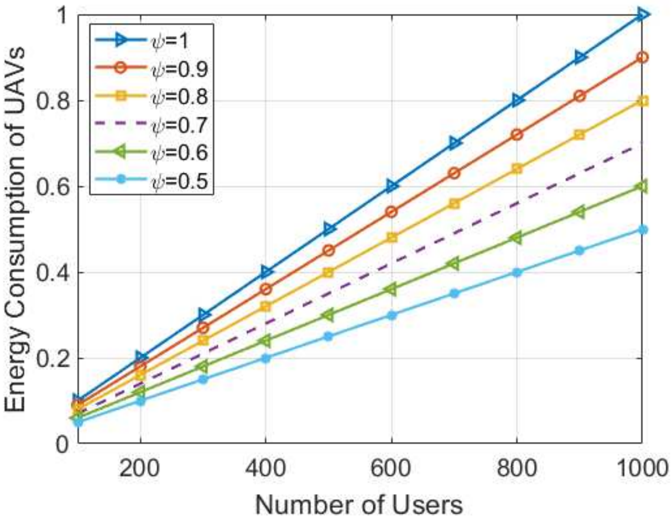


Figure 4.12: Normalized energy consumption of UAVs versus the number of outdoor users, where ψ illustrates the ratio of requests served by UAVs to the whole outdoor users' requests.

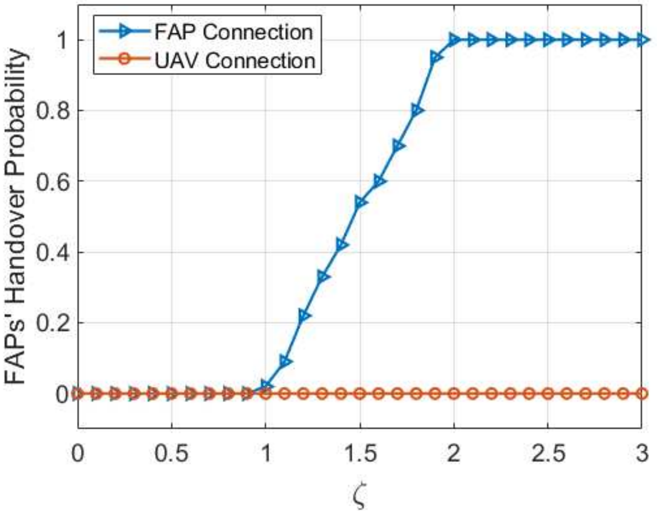


Figure 4.13: The FAPs' handover probability versus different values of $\zeta = \dfrac{\upsilon}{\upsilon_{\text{th}}}$.

can be seen from Fig. 4.13, serving HSUs ($\zeta > 1$) leads to triggering frequent handovers, where the handover probability is one for $\zeta > 2$, while there would not be any FAPs' handover by serving HSUs by UAVs.

4.2 Attention-based Coded/Uncoded Popularity Predictoin

In this Section, we developed the Multiple-model (hybrid) Transformer-based Edge Caching (MTEC) framework as a multi-content and time-series popularity prediction model. The MTEC framework captures both temporal and spatial correlation of multiple contents via multi-channel Transformer architectures, where the sequential request patterns of each content are given to a channel of the Transformer model. Our first objective for the development of the MTEC framework is to introduce a data-driven popularity prediction model with higher generalizability compared to existing works that predict the Top-K popular content relying on the historical request pattern. The second objective is to adapt the data-driven prediction model within the coded/uncoded content placement approaches. To achieve these objectives, the proposed MTEC framework is built upon the Transformer architecture, consisting of two parallel paths, which takes the historical request pattern of multiple contents as its input:

- The first path of the MTEC framework is a Transformer network, responsible for identifying the Top-K popular contents in the upcoming time, using the historical request pattern of contents. This part of the architecture is efficient for contents that their request patterns are smoothly changed over a short horizon.
- The second path of the MTEC framework is introduced to boost the generalizability of the learning model. In this path, for applicability to various types of content with different time-varying behavior, we relax the assumption of unchanged request patterns of content. Moreover, within the context of coded/uncoded content placement, we take one step forward and relax the assumption that the probability of content requests is known a-priori (i.e., Zipf distribution) [4]. In this regard, the second path is compromised of two stages, where the first stage is a Transformer network responsible for predicting the probability of requesting multiple contents in the upcoming time. The estimated request probability as the output of the first stage will be used for the coded/uncoded content placement to

identify which content should be stored partially/completely in the storage of caching nodes. Next, it will be concatenated with the historical request pattern (input of the first stage), provided as the input to the second stage, which analyzes the popularity of all contents simultaneously.

- The final output, which is the combination of the two parallel paths, is the Top-K popular contents in the upcoming time, which is applicable to various types of content with different time-varying behavior. The effectiveness of the proposed MTEC framework is evaluated through comprehensive studies on the real-trace multimedia request pattern, in terms of the classification accuracy, cache-hit ratio, and the transferred byte volume. Simulation results corroborate the effectiveness of the proposed MTEC framework in comparison to its counterparts over all the aforementioned aspects.

4.2.1 Multiple-Model Transformer-based Edge Caching (MTEC) Framework

The main goal of the proposed MTEC architecture is to predict the Top-K popular content using the historical request pattern of the underlying contents. In this context, we use MovieLens Dataset [212], where leaving a comment after watching a movie is considered a request [55, 200, 214].

In this Subsection, we present the constituent components of the proposed MTEC framework, where the main architecture is developed based on the Transformers (see Fig. 4.14). There are following drawbacks to the existing research works that motivate us to develop the MTEC framework:

i) With the assumption that users' preferences remain unchanged over a short horizon, existing works [5, 62, 63] predicted the Top-K popular content in the updating time t_{u+1}, where the input of the learning model was the request pattern of all contents in a time window with the length of L, ending at the updating time of t_u. This assumption works for content whose request pattern smoothly changes over time. It is, therefore, essential to

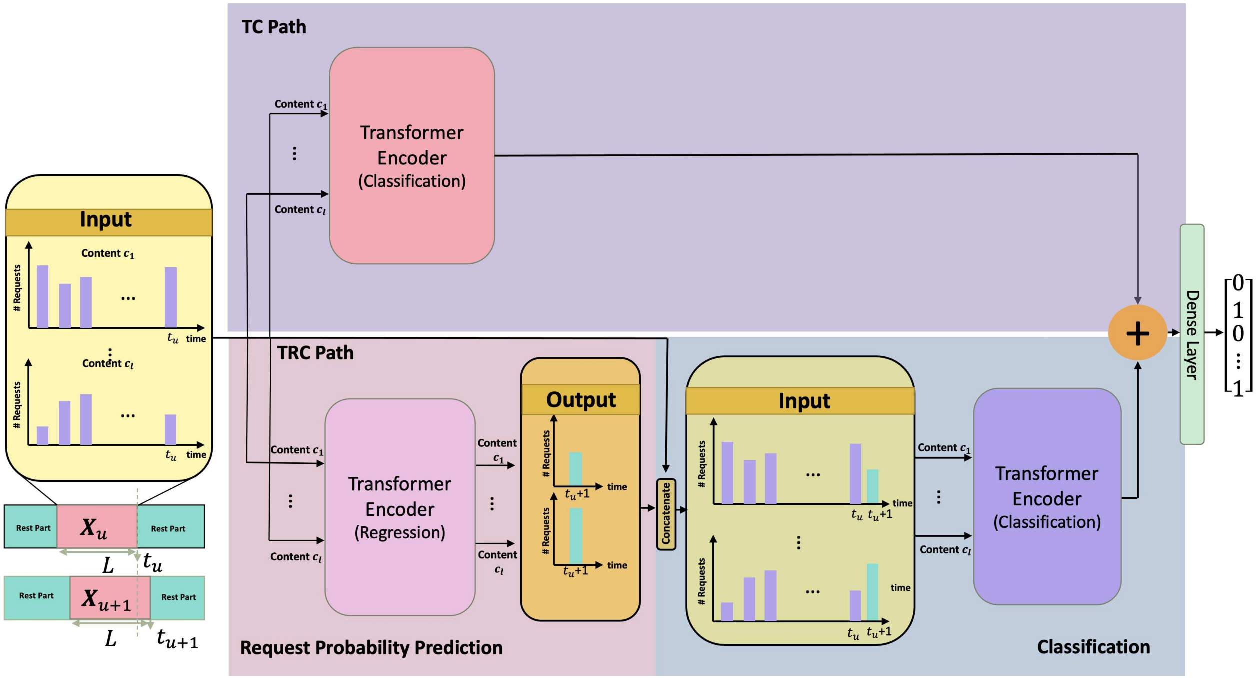


Figure 4.14: Block diagram of the proposed MTEC architecture.

develop a popularity prediction model with higher generalization ability, suitable for various types of content with different time-varying behavior.

ii) In a coded/uncoded content placement, the request probability of content in an upcoming time is also required to classify the Top-K popular content into two groups, i.e., popular and mediocre, while existing classification frameworks [5, 61, 63, 64] classified content as popular/non-popular.

To tackle the aforementioned challenges, we propose the MTEC framework, consisting of two parallel paths, where the first path is a Transformer-based Classification (TC) model, and the second path includes two series of Transformer-based blocks, named Transformer-based Regression and Classification (TRC) network. These are followed by a fully connected layer, as a fusion center combining the output of the two parallel paths to estimate the Top-K popular content. It should be noted that although the output of these two paths is similar in nature, simulation results illustrate that considering such an architecture improves the popularity prediction. Next, we introduce each of these blocks.

TC Path: This path is a multi-label classification model based on the Transformer model, where the input is the historical request pattern of multiple contents at time t_u and the output is the Top-K popular content at time t_{u+1}. The

Transformer is a type of Machine Learning (ML) model suitable for learning from sequential and time series data. Generally speaking, Transformers outperform the LSTM models because: (*i*) Although LSTM models are capable of learning long-term dependencies, they suffer from short-term memory over long sequences. Transformers, however, capture the connection/dependency between sequential components that are far from one another, resulting in higher accuracy; (*ii*) Due to the Multi-head Self-Attention (MSA) mechanism, Transformers can process data in parallel, reducing the training time; and, (*iii*) The attention mechanism of the Transformer eliminates the need to analyze data in the same order. Consequently, positional embedding is used to preserve the position information of an entity in sequential data.

Although this block is capable of capturing the request pattern of content with predictable behavior, i.e., with smooth changes over time in uncoded content placement, it would not be effective for content with sudden changes in a coded/uncoded manner. For these reasons, the second path is required.

TRC Path: The second path consists of the following two blocks:

i) **Request Probability Prediction Block:** The first block of the second path is used to predict the request probability of content at time t_{u+1} using the historical request pattern of content at time t_u (input data is the same as the first path). The output of this block will be used to classify the Top-K popular content (the final output of the MTEC framework) as the popular/mediocre one in the coded/uncoded content placement. Then, the Top-K popular content will be sorted in descending order, where $N_p = \lfloor \alpha C_f \rfloor$ and $N_a = N_s(C_f - \lfloor \alpha C_f \rfloor)$ are the cardinality of popular and mediocre content, respectively. Following our prior work [4], vector $\mathbf{z} = [z_1, \ldots, z_{N_a}]^T$ represents mediocre contents, where N_a denotes the cardinality of mediocre content. To identify which segments of mediocre content c_l, denoted by c_{ls} for $(1 \leq l \leq N_a)$ and $(1 \leq s \leq N_s)$, is cached in each FAP f_i for $(1 \leq i \leq N_b)$ belonging to an inter-cluster, an indicator matrix $\boldsymbol{Z}^{(f_i)}$ associated with FAP f_i is formed, where the l^{th} row of $\boldsymbol{Z}^{(f_i)}$, represented by $\boldsymbol{z}_l^{(f_i)} = [0, \ldots, 0, 1]_{(1 \times N_s)}$ is corresponding to the segments of file c_l cached in FAP f_i. Note that, $\sum\limits_{s=1}^{N_s} z_{ls}^{(f_i)} = 1$, where $z_{ls}^{(f_i)} = 1$, if the s^{th} segment of file c_l is cached in FAP f_i, otherwise, it would be zero. Therefore, the mediocre content of other FAPs f_j, for $(1 \leq j \leq N_b, j \neq i)$, belonging to an inter-cluster, is given by

$$\boldsymbol{z}_l^{(f_i)} \boldsymbol{z}_l^{(f_j)^T} = 0, \quad i = 1, \ldots, N_b, j = 1, \ldots, N_b, i \neq j. \tag{105}$$

Following the above discussion, different segments of mediocre content will be stored in an inter-cluster. Next, the same content as FAP f_i in one inter-cluster, is allocated to FAP f_k in the nearby inter-cluster, where k is given by

$$\boldsymbol{Z}^{(f_k)} = \boldsymbol{Z}^{(f_i)} \quad \text{if} \quad k = w^2 + wz + z^2, \tag{106}$$

where w is the number of FAPs required to move from FAP f_i in any direction, after which z number of FAPs should be moved by turning 60 degrees counterclockwise to reach FAP f_k [4] (see Fig. 4.2). Finally, the estimated request probability is appended to the original input samples to generate the input of the next block, which is used for the classification, i.e., identifying the Top-K popular content.

ii) **Classification Block:** In comparison to the CT block in the first path, the input of this block is both the historical request pattern of content at time t_u and the estimated one at t_{u+1}, resulting in higher classification accuracy for such content with sharp changes in their request pattern. Then, the output features of both paths are added, which are used as the input of the fusion layer (dense layer). The output of the dense layer is a vector $\mathbf{y}_u \in \mathbb{R}^{N_c}$, with K ones, where $1^{'}$s indicates the Top-K popular content (i.e., popular/mediocre one) and $0^{'}$s are non-popular content. Finally, the estimated Top-K popular contents are categorized into popular/mediocre one according to the output of the request probability prediction block.

Finally, we use Mean Squared Error (MSE) for the request probability prediction block, and binary cross-entropy as the loss function for the CT path, the classification block in the second path, and the fusion path. The overall loss function $\mathcal{L}$ of the proposed MTEC is given by

$$\mathcal{L} = w_1 \mathcal{L}_{RPP} + w_2 \mathcal{L}_{C_I} + w_3 \mathcal{L}_{C_{II}} + w_4 \mathcal{L}_F, \tag{107}$$

where $\mathcal{L}_{RPP}$, $\mathcal{L}_{C_I}$, $\mathcal{L}_{C_{II}}$, and $\mathcal{L}_F$ represent the loss function associated with the request probability prediction block, CT block, the classification in the second path, and the fusion path, respectively, and w_i, $i = \{1, \ldots, 4\}$ is the loss weight of each block.

4.2.2 Simulation Results

To evaluate the performance of the proposed MTEC framework, we consider a cluster-centric UAV-aided cellular network with $138,493$ GUs and $27,278$ number of distinct contents. Following the common assumption [131], we consider the scenario that the storage capacity of caching nodes is 10% of the total content, where the size of all multimedia contents is the same. Considering the GUs' location, determined based on their ZIP code [63], it is assumed that there are 21, and 3 terrestrial, and areal caching nodes, respectively, where each inter-cluster consists of $N_s = 7$ number of FAPs. Note that the classification accuracy is averaged over all caching nodes. To determine the best architecture of the multiple-model Transformer-based architecture, we first evaluate different versions of the proposed MTEC popularity prediction framework through trial and error. Moreover, the performance of a single classification model, i.e., Path 1 or Path 2, is evaluated, when they are trained independently. In all the experiments, the Adam optimizer is employed with learning rate of 0.0001 and weight decay of 0.00001. The activation function of the MLP layers in all Transformer models is ReLU, while it is sigmoid as the output layer. In classification blocks, the multi-content time-series request pattern data is converted to a sequential set of images, using the Gramian Angular Field (GAF) technique [129] to preserve the temporal correlations of the input data.

Effectiveness of the MTEC Architecture

This subsection evaluates the performance of the proposed MTEC architecture. Considering different hyperparameters, such as the number of heads, number of transformer layers, model dimension, and MLP size, we compare different variants of the proposed MTEC architecture in terms of accuracy and the number of parameters (complexity). As it can be seen in Table 4.4, increasing the

Table 4.4: Variants of the MTEC popularity prediction framework.

Model ID	**Layers (L)**	**Model dimension (d)**	**MLP layers**	**MLP size**	**Number of Heads**	**Params**	**Accuracy**
1	1	32	1	256	8	444,073	84.32 %
2	1	32	1	256	16	652,457	88.15 %
3	1	64	1	256	8	775,049	88.23 %
4	2	64	1	256	8	1,438,556	92.78 %
5	1	64	1	512	8	885,384	90.01 %
6	2	64	2	256	16	2,882,140	94.13 %

Table 4.5: The accuracy of the Top-K popular content using different window size (9, 49, and 99 days) for different variants of the proposed MTEC framework.

Model ID	**Accuracy**		
	9 Days	**49 Days**	**99 Days**
1	81.09 %	84.32 %	85.14 %
2	86.09 %	88.15 %	88.86 %
3	85.94 %	88.23 %	89.21 %
4	89.45 %	92.78 %	93.03 %
5	87.54 %	90.01 %	90.98 %
6	92.87 %	94.13 %	94.54 %

Table 4.6: The accuracy of the Top-K popular content using different loss weights.

Model ID	w_1	w_2	w_3	w_4	**Accuracy**
L1	0.2	0.4	0.1	0.3	94.13 %
L2	0.3	0.2	0.1	0.4	93.08 %
L3	0.0	0.0	0.0	1.0	90.54 %

number of heads from 8 to 16 (Models 1 and 2) and model dimension from 32 to 64 (Models 1 and 3) increase the accuracy of the proposed MTEC framework, while increasing the complexity of the learning model. Similarly, the classification accuracy is improved by increasing the number of transformer layers from 1 to 2 (see Models 1 and 4) and the MLP size from 256 to 512 (Models 3 and 5). According to the information provided in Table 5.9, the best architecture for the proposed MTEC framework is Model 6. It should be noted that although the number of parameters in Model 6 is higher than the others, the classification accuracy is higher as well. For that reason, we choose this model to compare it with other state-of-the-art.

We also evaluate the effect of window size, which is used for data segmentation

Table 4.7: Comparison between our methodology (MTEC framework) and other versions of the Transformer-based architectures.

Model Name	**Layers (L)**	**Model dimension (d)**	**MLP layers**	**MLP size**	**Number of Heads**	**Parameters**	**Accuracy**
TC	2	64	2	256	16	946,580	92.48 %
RCT	2	64	2	256	16	1,909,860	92.54 %
ViT	2	64	2	256	16	3,798,320	92.68 %
MTEC	2	64	2	256	16	2,882,140	94.13 %
MTEC-AT	2	64	2	256	16	2,882,396	93.28 %

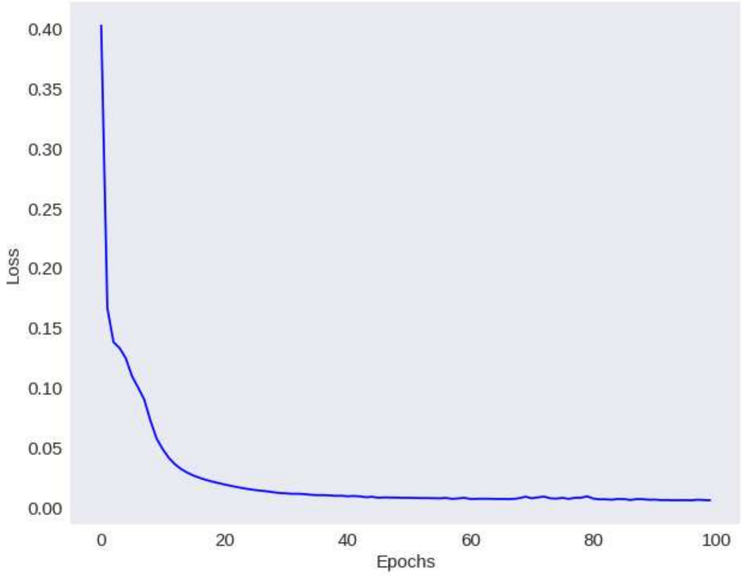


Figure 4.15: The convergence of the proposed MTEC framework.

during the dataset pre-processing phase. According to Table 4.5, we consider different window sizes (9, 49, and 99 days) to build the input samples $\mathcal{D} = \{(\mathbf{X}_u, \mathbf{y}_u)\}_{u=1}^{M}$. For instance, when the window size is 9, it means that we use the historical request pattern of content over the past 9 days to predict the popularity of content in the upcoming time. As shown in Table 4.5, extending the window length from 9 to 49 increases the classification accuracy, while there is no significant improvement considering very large window size, i.e, 99, due to the degradation in dependency of the number of requests with time.

We evaluate the effect of different loss functions on classification accuracy. According to Eq. (107), we consider different values for loss weight of each block, denoted by w_i, $i = \{1, \dots, 4\}$, to illustrate the effect of loss weight on the overall classification accuracy, shown in Table 4.6. Moreover, the convergence of the proposed MTEC framework is illustrated in Fig. 4.15. As shown in Fig. 4.15, increasing the number of epochs decreases the model loss, which shows that the model is well trained.

Moreover, to illustrate the superiority of the proposed MTEC architecture in comparison to a single Transformer, we compare it with different Transformer-based networks, such as the single TC model, corresponding to the first path of the MTEC model, the TRC network associated with the second path, the ViT model [5], and the MTEC-AT, which is the proposed MTEC architecture with a self-attention layer as the fusion layer instead of the fully connected one. According to Table 4.7, although the complexity of the proposed MTEC network is higher than other architectures, it provides higher accuracy.

Performance Comparisons

For comparison purposes, we applied seven state-of-the-art caching schemes to the Movielens dataset, including Least Recently Used (LRU) [51], Least Frequently Used (LFU) [51], PopCaching [214], LSTM-C [55], the TC scheme, the RCT scheme, and the TEDGE caching scheme [5], which is based on the ViT architecture. Fig. 4.16 compares the performance of the proposed uncoded MTEC scheme with other baselines listed above in terms of the cache-hit ratio, known as one of the widely used metrics in MEC networks. This metric indicates the number of requests managed by caching nodes versus the total requests made across the network. Note that, the proposed MTEC scheme can be used for both coded/uncoded content placement and the conventional uncoded one. Other baselines, however, are based on uncoded content placement, which is one of the main drawbacks of the existing data-driven caching schemes. In such a case that the multimedia content is partially stored in the storage of caching nodes, the cache-hit ratio would not be an accurate metric for the coded/uncoded content placement framework. For this reason and to be compatible with other state-of-the-arts, we first evaluate the performance of the proposed uncoded MTEC scheme in terms of the cache-hit ratio. Then, we define another metric suitable for the coded/uncoded content placement, known as the transferred byte volume, illustrating the ratio of the data volume, transmitted by caching nodes versus the total volume of the requested contents managed by caching nodes. As shown in Fig. 4.16, the optimal strategy [55] is defined as a caching scheme, where all requests through the network are served by caching nodes, which cannot be obtained in reality. According to the results in Fig. 4.16, the proposed MTEC caching framework achieves the highest cache-hit ratio in comparison to other state-of-the-art counterparts.

In terms of the transferred byte volume, it is assumed that $\alpha = 30$ percent of the storage of FAPs is associated with the popular contents, stored completely,

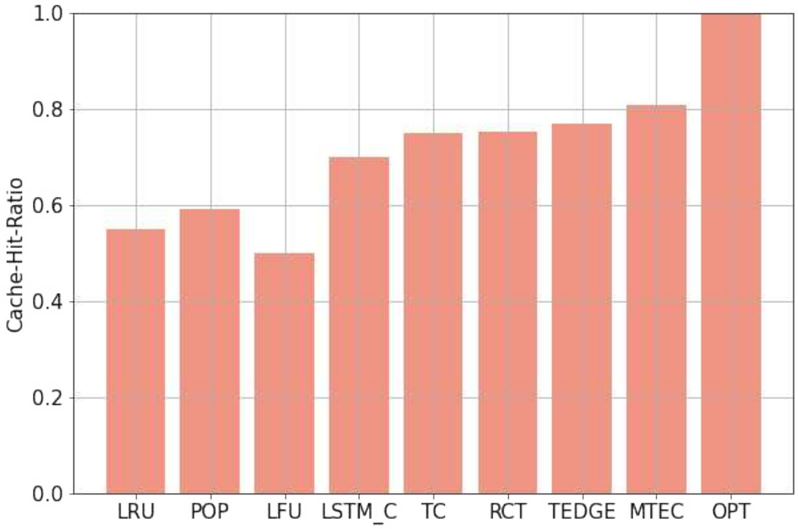


Figure 4.16: A comparison with state-of-the-arts based on the cache-hit ratio.

and 70% of the storage is assigned to mediocre content, stored partially according to the content placement strategy described in Section 5.5.1. As shown in Fig. 4.17, the byte volume transferred by caching nodes in the proposed MTEC framework is higher than other counterparts. In comparison to the Cluster-centric and Coded UAV-aided Femtocaching (CCUF) [4] framework, the coded/uncoded content placement in the CCUF is performed based on the historical request probability of content, while the proposed MTEC and RCT frameworks use the predicted one. Moreover, since the classification accuracy of the RCT model is lower than the proposed MTEC architecture, the MTEC framework outperforms in terms of the transferred byte volume.

4.3 Conclusion

In this Chapter, we developed the CCUF framework in Section 4.1 for an integrated and dynamic cellular network to maximize the number of requests served by caching nodes. To increase the cache diversity and to store distinct segments of contents in neighboring FAPs, we employed a two-phase clustering technique for FAPs' formation and UAVs' deployment. In this case, we formulated the success probability of the proposed CCUF framework. Moreover, in the cluster-centric cellular network, multimedia contents were coded based on

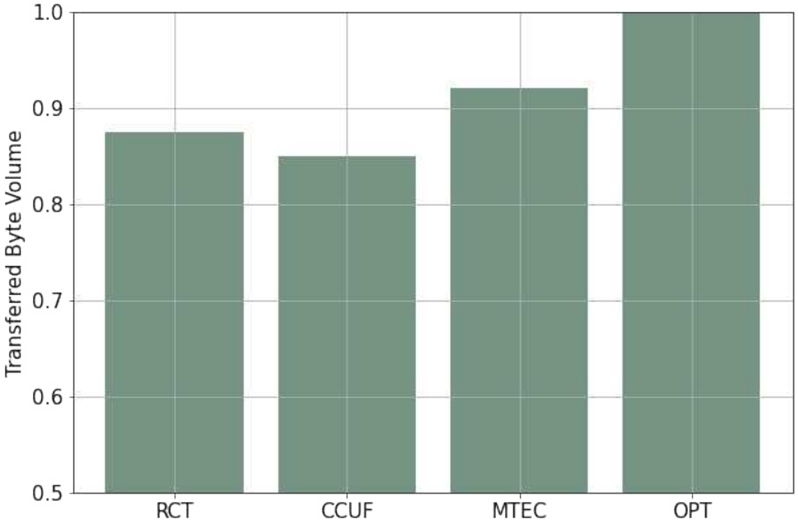


Figure 4.17: A comparison with state-of-the-arts based on the transferred byte volume.

their popularity profiles. In order to benefit the CoMP technology and to improve the inter-cell interference, we determined the best value of the number of contents that should be stored completely. According to the simulation results and by considering the best value of α, the proposed CCUF framework results in an increase in the cache-hit-ratio, SINR, and cache diversity and decrease users' access delay and cache redundancy. Going forward, several directions deserve further investigation. First, it is of interest to introduce a RL-based method for outdoor environment, where ground users can be autonomously served by UAVs or FAPs, based on the dynamic population of their current locations and their speeds. Second, the optimum number of ground users to be served by a UAV in the proposed network needs to be analyzed.

Moreover, in Section 4.2, we presented an efficient multi-content time-series popularity prediction model referred to as MTEC framework with the application to the cluster-centric MEC networks. Due to the lack of predicted request probability, existing data-driven caching strategies were inefficient for coded/uncoded content placement. To tackle this issue, we developed a multiple-model Transformer-based architecture to not only predict the upcoming Top-K popular content but also estimate the request pattern of multiple contents simultaneously, which was used to determine which contents should be stored partially or completely. Simulation results showed that the proposed MTEC caching scheme improves the cache-hit ratio and the transferred byte volume when compared to its state-of-the-art counterparts.

Chapter 5

Indoor D2D-enabled MEC Network

With the focus on the integrated indoor/outdoor transmission schemes, introduced by the CCUF framework in the previous Chapter, it is required to have an efficient indoor caching strategy. In this chapter, an indoor macro cellular network is considered consisting of FAPs, which are equipped with extended storage, and Mobile Users (MUs) that are capable of communicating with each other via D2D communication. In such a network with a femtocell infrastructure, we have unlimited-energy caching nodes, while we benefit from the D2D communication through increasing the coverage of the network. Within this context, our goal is to overcome the following two key challenges:

(i) In a dynamic femtocaching network where users move consistently in the coverage area of the caching nodes, it is not possible to download completely a requested content in one contact. Therefore, it is supposed that all contents are partially saved into the storage of caching nodes, and;

(ii) Deployment of massively dense FAPs with small transmission ranges in an enterprise wireless network leads to frequent handovers, particularly for high speed users, since they connect to a specific FAP for a short time.

To address the aforementioned challenges, in summary, we propose an indoor femtocaching scheme referred to as the Mobility-Aware Femtocaching algorithm based on Handover (MAFH), in which, the best caching node is selected according to the RSSI value and the velocity of users as decision criteria. In order to reduce the number of unnecessary handovers, the appropriate target FAP for handover is chosen in such a way that the time interval between successive handover triggers will be extended, which consequently reduces the number of handovers during a connection.

Focusing on the indoor caching strategy, we need to have an indoor localization framework to precisely localize users within indoor areas to determine which access point should manage the MUs' requests. Therefore, in this Chapter we propose two indoor localization frameworks to be used in the indoor D2D-enabled MEC networks. More precisely, the unreliability of indoor localization frameworks in the presence of the multi-path effect, is a key limitation which is an unavoidable factor. By considering the effects of multi-path and path-loss on the train dataset, the need for complex and precise analytical models can be eliminated. For this reason, we then proposed an efficient CNN-based BLE indoor localization framework, where AoA information is used as the input of the CNN. We considered indoor environments without presence of LoS links affected by AWGN with different SNRs and Rayleigh fading channel. Moreover, by assuming a 3D indoor environment, the destructive effect of the elevation angle of the incident signal is considered on the position estimation.

Moreover, UWB technology is another potential candidate for providing reliable, accurate, and energy-efficient indoor navigation/localization systems. Although UWB technology can enhance the accuracy of indoor positioning due to the use of a wide-frequency spectrum, there are key challenges ahead for its efficient implementation. On the one hand, achieving high precision in positioning relies on the identification/mitigation NLoS links, leading to a significant increase in the complexity of the localization framework. On the other hand, UWB beacons have a limited battery life, which is especially problematic in practical circumstances with certain beacons located in strategic positions. To address these challenges, we, then, introduced an efficient node selection framework to enhance the location accuracy without using complex NLoS mitigation methods, while maintaining a balance between the remaining battery life of UWB beacons. Referred to as the Deep Q-Learning Energy-optimized LoS/NLoS

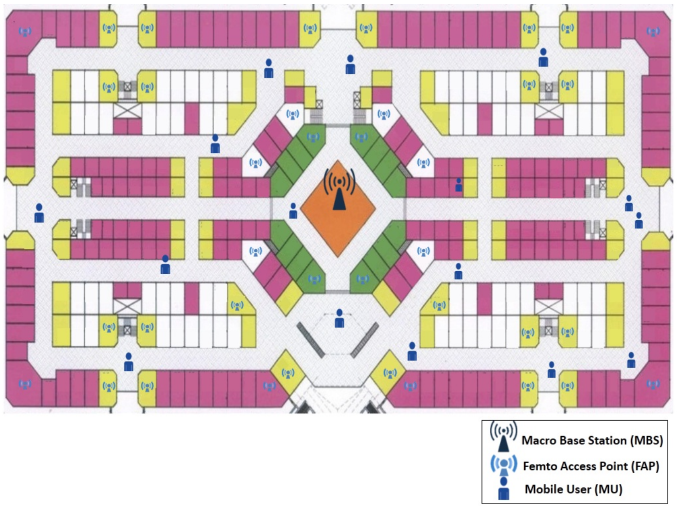


Figure 5.1: The map of one typical enterprise femtocell network consisting of a Macro Base Station (MBS), Femto Access Points (FAPs), and some Mobile Users (MUs).

(DQLEL) UWB node selection framework, the mobile user is autonomously trained to determine the optimal pair of UWB beacons to be localized based on the 2D TDoA framework.

5.1 Connection Scheduling in MAFH Framework

In this Section, we introduce the Mobility-Aware Femtocaching algorithm based on Handover (MAFH) framework, in which, the best caching node is selected according to the RSSI value and the velocity of users as decision criteria. We consider a practical macro cellular network is considered consisting of FAPs, which are equipped with extended storage, and Mobile Users (MUs) that are capable of communicating with each other via D2D communication. In such a network with a femtocell infrastructure, we have unlimited-energy caching nodes, while we benefit from the D2D communication through increasing the coverage of the network.

5.1.1 Problem Description and Assumptions

We consider an enterprise femtocell network, as shown in Fig. 5.1, consisting of a macro base station, denoted by MBS, located at the center of market with the communication range R and N FAPs, denoted by f_i, $i = 1, \dots, N$, with equal cache size, denoted by C_f, and an equal transmission range r_f. We assume multiple MUs in the network, denoted by u_j, $j = 1, ..., S$, moving based on the random walk model as described shortly. Furthermore, all MUs are equipped with the storage of fixed sizes C_u such as those available on modern smart phones, and a fixed transmission range r_u. The storage capacity of mobile users C_u is much more limited than the amount of C_f. The MUs request for a file from a library of M contents $\mathcal{C} = \{c_1, \ldots, c_M\}$, where the video file c_k has the size s_{c_k}. It should be noted that the MBS can completely cover the enterprise area and serve all S MUs in the network. Nevertheless, after requesting a content by a mobile user, this request is immediately served by FAPs or nearby clients in order to reduce the traffic load on the MBS.

In this work, we assume that each FAP f_i operates in the open access mode, in which any MU u_j, existing in the transmission range of f_i is allowed to access to the femtocell. Therefore, we utilize the TDoA localization method to estimate the current location of MUs in the vicinity of each FAP. Also, it is supposed that mobile users are able to establish D2D communication, which is controlled by the MBS. As a result, a central control unit is aware of the cached contents of all users.

Taking the above considerations into account, the main objective of this work is to propose a mobility-aware femtocaching strategy based on a real dense network, including FAPs and user devices, supported by D2D communications. Therefore, when a user requests a content, there are different alternative ways to serve this request, which will be fully described in “User Access Pattern" Subsubsection. To be more practical, it is also assumed that users with different velocities move in all directions in the network leading to occurrence of handover during a request, that will be introduced in “User Mobility Pattern” and “Handover Phenomenon” Subsubsections, respectively. Moreover, transmission of large-sized video contents in one contact is not feasible, especially in such a case that mobile users move across the transmission range of different FAPs rapidly. Therefore, all contents are segmented based on the Linear Random Fountain Code method that will be introduced in “Rateless Code” Subsubsection. To handle this problem, we propose a mobility-aware femtocaching strategy based on handover to select the best caching node to manage users’ requests. Toward this goal, we briefly introduce some concepts required for the proposed strategy as follows:

User Mobility Pattern

In this work, we assume that mobile users are able to move in all directions instead of just moving forward along a straight line in corridors as well as inside offices of the market place. To illustrate this movement pattern, the Difference Correlated Random Walk (DCRW) model is considered. A random walk is defined as a random process where the current position of mobile user u_j at time t, denoted by L_j^t, depends on the velocity of movement. There are numerous indoor speed estimation methods, including E-eyes [171], and CSI-based human Activity Recognition and Monitoring (CARM) [172] that utilize some prominent features of Channel State Information (CSI) for speed recognition. In this work, the velocity of user u_j at time t in two dimensions, denoted by $\boldsymbol{v}_j^t$, is calculated by solving bivariate Stochastic Differential Equation (SDE) as follows [173]:

$$d\boldsymbol{v}_j^t = -\begin{pmatrix} -\log \gamma_1 & \theta \\ -\theta & -\log \gamma_2 \end{pmatrix} (\boldsymbol{v}_j^t - \boldsymbol{\mu})dt + \mathbf{J}d\mathbf{B}_t, \tag{108}$$

where γ_1 and γ_2 represent auto-correlation parameters in the first and second coordinates, $\boldsymbol{\mu}$ and θ denote the vector of mean velocity and the mean turning angle of the DCRW model, respectively. Moreover, $\mathbf{J}$ is an 2×2 lower triangular matrix with positive diagonal elements, which determines the covariance of the changes in velocity, and $\mathbf{B}_t$, as 2×1 vector, denotes standard Brownian motion at time t. By discretizing the SDE, the location $\mathbf{L}_j^t$ of user u_j is obtained as follows [173]:

$$\mathbf{L}_j^t = \mathbf{L}_j^{t-1} + \boldsymbol{v}_j^{t-1}\Delta t, \tag{109}$$

where Δt denotes the time interval between two estimated locations at time t and $t-1$. As seen in Fig. 5.2, all offices in the market place are equipped with a FAP, and the distribution of FAPs is dense enough to overlap with each other. Therefore, when a mobile user requests a specific content and moves straight along the corridor with a constant velocity at the same time, it passes through

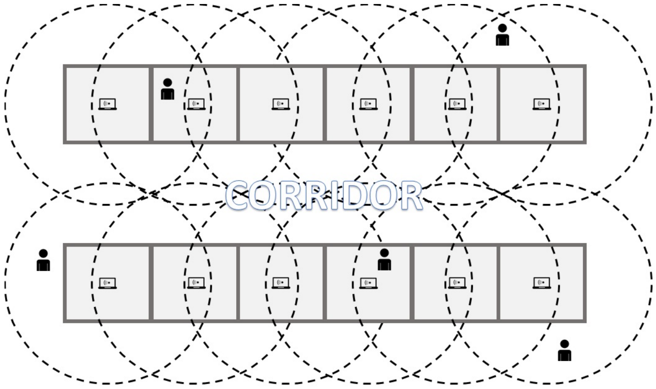


Figure 5.2: A typical structure of the enterprise femtocell network consisting of 12 shops located at both sides of the corridor.

the coverage area of various femtocells and experiences several consecutive handovers during a request. In this regard, mobile users are categorized, according to their speeds, into two distinguished groups as follows:

- **Low Speed Users:** The mobile users in this category have no apparent motion in their places, e.g., user u_j who works with a device that is in the same location for many hours, or just move slightly in the coverage area of a particular FAP f_i with speed υ_j^t, where υ_j^t is less than a predefined threshold υ_{th}.
- **High Speed Users:** In this category of clients, the user's speed υ_j^t is more than or equal to υ_{th}.

Rateless Code

In a dynamic femtocaching system where MUs move across the coverage area of caching nodes during a specific request, it is fairly unlikely that the user's request is served by a particular caching node in one contact. For instance, let us consider an illustrative example consisting of a small wireless network where some MUs move within the coverage area of two distinct FAPs f_1 and f_2. We also suppose that there are two equally popular files c_1 and c_2, and the optimal caching policy would store a complete copy of either file c_1 or file c_2 in the cache of f_1 and f_2. As long as file c_1 is stored in the cache of two FAPs, all users can successfully download this file by the two encountered FAPs; i.e., half unit of data can be received by each FAP. However, all the requests for file c_2 will be managed by the MBS. Whilst, for this case, where users move and access pairs of FAPs, the optimal caching policy changes to store the half unit of each file in the cache of FAPs f_1 and f_2. In this case, each FAP stores parts of the file that can be combined by the client to reconstruct the requested file. Hence, none of the requests will be redirected to the MBS. This example shows the inefficiency of caching schemes that neglect the user mobility and contact duration limits (e.g., [174]). For this reason, we assume that contents are saved partially as a result of the Fountain code which is inherently a rateless code.

In this work, we deploy an efficient two-phased encoder, named Linear Random Fountain Code (LRFC), in which each video file c_k with the size s_{c_k} fragments into T input symbols and consists of $c_{k,l}$ encoded segments with the same size s_{c_k}, in which $k \in \{1, \ldots, M\}$ and $l \in \{1, \ldots, T\}$ [175]. In the placement phase, each FAP stores δ_{c_k} of output symbols from file c_k. It should be noted that the same number of output symbols of file c_k is stored in the storage of FAPs, however, the sets of $c_{k,l}$, $l \in \{1, \ldots, T\}$, which is cached in the storage of various FAPs are different. Moreover, since clients are equipped with a storage, in which C_u is much more limited than the amount of C_f, after complete downloading of content c_k, requesting user stores σ_{c_k} of output symbols from file c_k, where the value of σ_{c_k} is less than δ_{c_k}. Accordingly, during the delivery phase, file c_k can be recovered by collecting K_{c_k} encoded segments, where the value of K_{c_k} depends on the size of file c_k. We also assume that the duration of each contact between user u_j and caching node, denoted by n_q, including $u_{j'}$ and FAP f_i, is $t^c_{j,q}$ time slots, and the transmission rate from caching node n_q to u_j is $r_{j,q}$ bps. As a result, the $B_{j,q} = \lfloor \frac{t^c_{j,q} r_{j,q}}{s_{c_{k,l}}} \rfloor$ segments can be transmitted during one contact from n_q to u_j, where $\lfloor \rfloor$ denotes the floor operator. The LRFC encoder generates a sequence of output symbols $c_k = \{c_{k,l}, \ldots, c_{k,\varepsilon}\}$, where ε represents the number of outputs symbols. In this case, m^{th} output symbol of file c_k is obtained as follows:

$$c_{k,m} = \sum_{a=1}^{T} g_{a,m} c_{k,a}, \tag{110}$$

where $g_{a,m}$ represents an independent coefficient, which is generated randomly with uniform distribution. For fixed ε, parameter c_k can be obtained as follows:

$$c_k = \mathbf{c_k G}, \tag{111}$$

where $\mathbf{c_k}$ and $\mathbf{G}$ are the vector of input symbols and $T \times \varepsilon$ matrix, respectively. It should be noted that when a MU requests content c_k, it is necessary to collect a set of K_{c_k} output symbols $\mathbf{y} = (y_1, \ldots, y_{K_{c_k}})$ to decode a file completely, in which $\mathbf{y}$ is expressed as follows:

$$\mathbf{y} = \mathbf{c_k} \tilde{\mathbf{G}}, \tag{112}$$

where $\tilde{\mathbf{G}}$ indicates a matrix corresponding to the K_{c_k} columns of $\mathbf{G}$, associated with the collected output symbols. Totally, if $\tilde{\mathbf{G}}$ is full rank, the decoding process would be performed successfully after recovering $\mathbf{c_k}$, otherwise, the receiver needs more additional output symbols.

User Access Pattern

Generally, the highlighted aim of a femtocaching-based network is to reduce the number of satisfied requests served by the MBS. Due to the mobility of users, we define the delay constraint, denoted by T_d, as the time that users are allowed to collect segments of required contents through one of the caching nodes as described shortly, otherwise, this request will be managed by the MBS. In this regard, when the file $c_k \in \mathcal{C}$ is requested by user u_j, this request can be served through the following four methods:

- **Local Cache:** Firstly, if requested content c_k exists in the cache of own user u_j, this user would obtain the required content c_k from the local cache, otherwise, user u_j downloads the required content through a D2D cache or FAP cache existed in the transmission range of user u_j within the deadline T_d.
- **FAP Cache:** When the file c_k is requested by user u_j exited in the communication range of FAP f_i, regardless the speed of user, this request is handled by f_i immediately. In the first case, if c_k is found in the cache of f_i, it is directly sent to the client by FAP and the cache hit occurs, otherwise,

this request is managed by the MBS, and the cache miss happens. In such a case that multiple MUs with the same velocity are located in the transmission range of FAP f_i and request different files simultaneously, where all contents exist in the cache of the corresponding FAP, these users are ranked as follows [131]:

$$e_l = \frac{N_l}{T_l}, \qquad l = 1, \ldots, S_i, \tag{113}$$

where N_l and T_l represent the number of satisfied requests of user u_l, managed by caching nodes and the total number of user's requests u_l, respectively. Moreover, S_i denotes the number of users with the same velocity that should be served by FAP f_i. A small value of e_l indicates the minority of user's requests served by FAP. Therefore, the priority of this user is more to be managed by FAP in order to experience less latency.

- **D2D Cache:** According to the user's speed, if u_j is a low speed user and another mobile device, storing the requested content in the cache, exists within the transmission range r_u, the D2D communication will be established between two users. It should be noted that since high speed users move rapidly through corridors, deploying the D2D communication is not feasible for this type of users.
- **MBS:** Eventually, if the user cannot access at least K_{c_k} different encoded segments of file c_k through one of the above-mentioned methods within the deadline T_d, this request will be managed by the MBS to obtain the remaining segments. Moreover, in such a case that u_j locates in the transmission range of FAP f_i or there are other MUs in the vicinity of u_j to establish D2D communication, but file c_k cannot be found in their storage, u_j must be served by the MBS. In this work, we assume that the D2D communication and the FAP communication do not interfere with the communication between the MBS and users. This assumption is justified if the D2D and femtocell communications occur in separate frequency bands (e.g., WiFi). In addition, it is assumed that the MBS assigns different frequency sub-channels to D2D users and FAPs by employing an efficient resource scheduler to avoid the co-channel interference [40].

Handover Phenomenon

By considering the fact that clients existed in the proposed network can move in corridors and shops and due to the small transmission range of FAPs, mobile users with a high velocity presumably experience unnecessary handovers during a request. To tackle this problem, we first define the pathloss between FAP f_i and client u_j denoted by $PL_{i,j}(dB)$ as [176]:

$$PL_{i,j}(dB) = \mathtt{max} \begin{Bmatrix} 38.46 + 20\log_{10}(d_{i,j}) + L_{ow} + X_\sigma \\ 15.3 + 37.6\log_{10}(d_{i,j}) + L_{ow} + X_\sigma \end{Bmatrix}, \tag{114}$$

where $d_{i,j}$ denotes the distance between FAP f_i and mobile user u_j, and L_{ow} represents the exterior wall penetration loss of shops in the network area, which is 10 dB or 20 dB. Moreover, X_σ is a zero-mean Gaussian-distributed random variable with standard deviation σ that represents the effect of multi-path fading in our femtocaching network [176]. For the above pathloss description, the strength of the signal received from FAP f_i by user u_j, denoted by $RSSI_{i,j}$, is obtained as

$$RSSI_{i,j} = P_i - PL_{i,j}, \tag{115}$$

where P_i is the transmission power of FAP f_i. We also define the threshold level P_{th} as the minimum signal strength that one MU can receive with a high quality. In traditional handover algorithms (e.g., [177]), when the mobile user u_j is moving away from the area covered by FAP f_i, once $RSSI_{i,j}$ drops below the threshold level P_{th}, handover is occurred, and then u_j starts to find the strongest signal among neighboring candidate FAPs as a target for handover which satisfies the following condition:

$$RSSI_{i',j} = RSSI_{i,j} + hm, \tag{116}$$

where $RSSI_{i',j}$ and hm represent the RSSI from the target FAP $f_{i'}$ and the hysteresis margin, respectively, which is a commonly used parameter for avoiding frequent handovers. Since the transmission ranges of FAPs are small, during movement of a MU, the RSSI drops below the threshold level P_{th} again in a short time interval and another handover is occurred. As a result, the number of handovers in the network increases dramatically, especially, when a client moves rapidly among femtocells. To tackle this problem, in the next Subsection, we propose an efficient mobility-aware femtocaching strategy to minimize the number of unnecessary handovers.

5.1.2 Proposed MAFH Framework

The users' mobility across the wireless network leads to the handover phenomenon, when a contact is in progress between u_l and f_i and u_l leaves the coverage area of f_i. As a result, user u_l explores another FAP existed surrounding to connect and download the remaining parts of the content. Therefore, a large number of handovers is expected to occur during one contact, especially when MUs move with a high velocity between femtocells. This issue motivates us to propose an efficient strategy, namely Mobility-Aware Femtocaching scheme based on Handover (MAFH) for the network's model introduced in Subsection 5.1.1. In the proposed system model, the caching nodes' deployment, including FAPs and other mobile devices equipped with the storage, is dense enough, therefore, when a user requests a content, there are more than one candidate to connect. For this reason, it is crucial to prioritize caching nodes in order to improve the connecting time, handover occurrence, and the energy consumption. In the next subsection, we introduce some common performance metrics to evaluate the proposed MAFH scheme.

Performance Metrics

Cache Hit Ratio: This is a common performance metric in caching topic which is defined as the ratio of all users' requests served by caching nodes versus total requests. After requesting content c_k by user u_l, if at least one caching node in the vicinity of user u_l includes file c_k, this request would be immediately served by the corresponding caching node. In this case, the cache hit occurs, otherwise, this request would be served by the MBS and the cache miss happens. The cache hit ratio, denoted by $\mathcal{CH}$, is calculated as follow:

$$\mathcal{CH} = \frac{n_T - n_{MBS}}{n_T}, \tag{117}$$

where n_T and n_{MBS} denote the total number of requested segments in the network, and the number of unsatisfied requests, directly managed by the MBS.

Transferred Byte Volume: This metric is used to express the amount of data flowing directly from the MBS in order to satisfy users' requests. Without loss of generality, it is assumed that the size of all contents are the same. Therefore, the total volume of contents requested in the network, would be expressed as $\mathcal{S}_T = n_T s_{c_{k,l}}$. Similarly, the volume of contents served by MBS is $\mathcal{S}_{MBS} = n_{MBS} s_{c_{k,l}}$. In this case, the transferred byte volume, denoted by $\mathcal{TB}$, is obtained as

$$\mathcal{TB} = \frac{\mathcal{S}_{MBS}}{\mathcal{S}_T}. \tag{118}$$

Note that an efficient cache policy that maximizes the number of cache hits, leads to a higher cache hit ratio, lower transferred byte volume, and better resource utilization.

Connecting Time: This metric is defined as the time interval that mobile user u_j can stay connected to the current caching node n_q, denoted by $\tau_{j,q}$, for downloading the segment $c_{k,l}$, calculated as follow:

$$\tau_{j,q} = \frac{\Delta x_j}{\upsilon_j^t}, \tag{119}$$

where Δx_j represents the distance that user u_j moves during $\tau_{j,q}$ while it connects to caching node n_q, and υ_j^t denotes the velocity of movement of MU u_j. In this case, the normalized connecting time, denoted by $\mathcal{CT}$ is given by

$$\mathcal{CT} = \frac{\tau_{j,q}}{t_{j,q}^c}, \tag{120}$$

where $t_{j,q}^c$ denotes the required time duration of contact between user u_j and caching node n_q to completely download $c_{k,l}$.

Handover Rate: This metric indicates the number of handovers per minutes.

Handover frequency is typically increased by increasing the speed of mobile users, and the number of FAPs, installed in the network. In this context, the most effective femtocaching strategy is that mobile users connect to the further FAP instead of close to one in order to reduce the handover rate.

Energy Consumption: The energy consumption of user u_q in the transmission mode in D2D communication when this user transmits segment $c_{k,l}$ to user u_j is calculated as follow [178]:

$$E_{q,j}(c_{ls}) = \frac{P_{q,t} s_{c_{k,l}}}{R_{q,j}}, \tag{121}$$

where $P_{q,t}$ denotes the power consumption of user u_q in the transmitter state, and $R_{q,j}$ represents the channel capacity, given by

$$R_{q,j} = W_q \log_2(1 + \frac{K P_{q,t}}{N_0 W_q d_{q,j}^2}), \tag{122}$$

where W_q and K represent the transmission bandwidth, and pathloss factor, respectively. Moreover, N_0 and $d_{q,j}$ denote the noise power spectral density, and the distance between users u_q and u_j.

Delivery Time: This metric illustrates the time duration $\mathcal{T}_d$ that takes to download one segment $c_{k,l}$ of the required content by user u_j from caching node n_q. $\mathcal{T}_d$ is calculated as follow:

$$\mathcal{T}_d = \frac{s_{c_{k,l}}}{r_{q,j}} + \frac{d_{q,j}}{c}, \tag{123}$$

where $c = 3 \times 10^8$ denotes the speed of light.

Methodology

To implement a dynamic mobile network, in which users are moving in all directions, we assume the random walk model to describe how users move through the network. In this regard, all the following steps are performed for the MAFH scheme:

Step 1: Caching Nodes Identification: After requesting a content, this

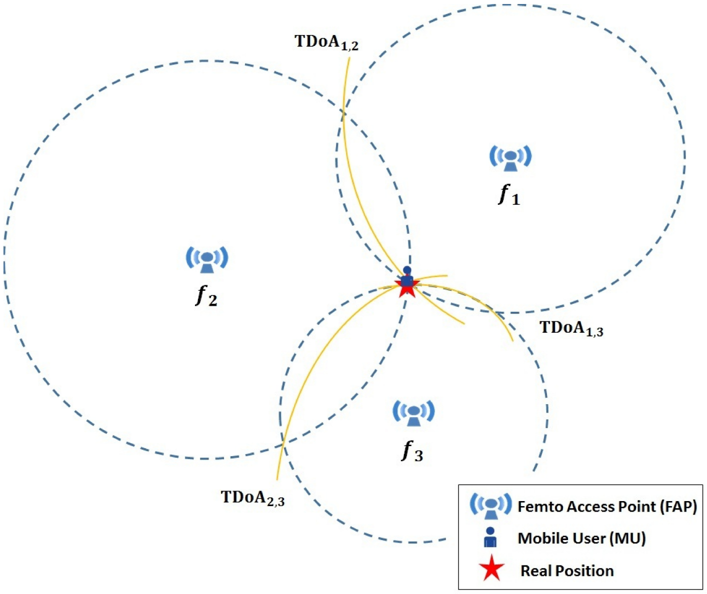


Figure 5.3: MU's location estimation based on the TDoA method.

request must be served by one of the neighboring caching nodes. Therefore, in the first step, all users need to scan nearby caching nodes, including FAPs and other mobile devices. For an enterprise femtocell network like the market place shown in Fig 5.1, and to recognize the exact location of users for service, we need to deploy an efficient indoor localization method. Among various signal metrics for indoor localization, including RSSI, Channel State Information (CSI), and TDoA, we measure the distance between different caching nodes by computing TDoA. Since wireless signals become weaker in proportion to the distance, path-loss model is used to estimate the location of nodes. As a result of indoor obstacles, such as walls and human beings, a transmitted signal can be reflected, refracted and diffracted, and a number of phase delayed and power attenuated versions of the same signal is received by users. Therefore, it is essential to deploy a highly accurate, easily implemented, and cost-effective localization method, which is robust against multipath fading and indoor noise. As shown in Fig. 5.3, in the TDoA method, the time difference of received signals from at least three FAPs is required to estimate the exact location of MUs. Toward this goal, it is assumed that $\mathbf{L}^l_{u,t} = [x^l_{u,t}, y^l_{u,t}]^T$ and $\mathbf{L}^i_f = [x^i_f, y^i_f]^T$ denote the location of user u_l at time t and FAP f_i, respectively. Based on the assumption that the first FAP is the reference node, the TDoA measurement between FAP 1 and FAP i is obtained as follows:

$$t_{i,1} = t^{'}_{i,1} + \Delta t_{i,1}, \quad i = 2, \ldots, N, \tag{124}$$

where $\Delta \mathbf{t} = [\Delta t_{2,1}, ..., \Delta t_{N,1}]^T$ represents an error vector containing zero-mean Gaussian random variables, and $t^{'}_{i,1}$ denotes the real time difference of received signals. By considering the signal propagation speed, denoted by c, the TDoA measurement is related to the Range Difference of Arrival (RDoA) as follows:

$$r^{'}_{i,1} = ct^{'}_{i,1}, \quad i = 2, \ldots, N, \tag{125}$$

where $r^{'}_{i,1}$ represents the actual distance between FAP 1 and FAP i, and $r_{i,1}$ is obtained as follows:

$$r_{i,1} = r^{'}_{i,1} + c\Delta t_{i,1}, \quad i = 2, \ldots, N. \tag{126}$$

Moreover, to calculate $r^{'}_{i,1}$, we have:

$$r^{'}_{i,1} = r^{'}_{i} - r^{'}_{1} = \|\mathbf{L}^{l}_{u,t} - \mathbf{L}^{i}_{f}\| - \|\mathbf{L}^{l}_{u,t} - \mathbf{L}^{1}_{f}\|, \tag{127}$$

where $r^{'}_{i} = \|\mathbf{L}^{l}_{u,t} - \mathbf{L}^{i}_{f}\|$ denotes the true distance between user u_l and FAP f_i. Consequently, the exact location of MUs are determined.

Step 2: Speed Estimation: Since the velocity of MUs is the key factor in selecting the best type of connection, the SDE method, described in Subsection 5.1.1, is deployed to track continuously the unstable velocity of users.

Step 3: Connection Type: Regarding the users' speed, the best caching node to serve users' requests is determined as the result of one of the following cases:

Case 1 (High Speed Users): In the case of high speed users, it is not beneficial to establish the D2D communication because the connection between mobile devices cannot sustain long enough to avoid frequent handover to other devices. Therefore, high speed users should only connect with FAPs that is a challenge for fast moving users. In our proposed algorithm, the distance between mobile device and FAP is used as a metric to select the candidate FAP. More specifically, we select a farther FAP with remaining file segments in its storage for handover to achieve long uninterrupted connection instead of near FAP. At the time of handover decision, the RSSI from a farther FAP is less than the RSSI from a nearby FAP. However, in a short range communication model, as the MU moves straight between femtocells, the distance between the MU and the target FAP decreases, consequently, the RSSI from the target FAP increases. Thus, by choosing such a farther femtocell as the target, the MU can stay connected with that FAP for a longer time, which extends the time between two consecutive handovers. To attain this goal, the RSSI from all available FAPs in the vicinity of a user along with FAPs' IDs are recorded in Neighbor Femtocells List (NFL). This list is updated every time the user requests a content, such as

$$NFL = \{f_j, j = 1, \ldots, n | RSSI_{j,l} > P_{th}\}, \tag{128}$$

where $RSSI_{j,l}$ indicates the RSSI level of FAP f_j at the user u_l which is greater than or equal to P_{th}, and $n < N$ is a random variable denoting the number of available FAPs surrounding the user u_l. It is possible that the case $j = 0$ occurs when there is no available FAP for the subsequent handover. The NFL is used to determine which femtocell has been visited by user u_l moving through the corridor before occurring handover. When a handover process is invoked, a new set of RSSI values from the available femtocells along with the FAPs' IDs are recorded in a Candidate List (CL), such as

$$CL = \{f_{j'}, j^{'} = 1, ..., n^{'} | RSSI_{j',l} > P_{th}\}, \tag{129}$$

where $RSSI_{j',l} \neq RSSI_{j,l}$ and $n^{'} < N$ is a random variable denoting the number of candidate femtocells for handover. It is possible that the NFL set may be empty, while the cardinality of the CL set is at least one, as the CL is typically formed sometime after the NFL.

To clarify the beneficial of the proposed strategy, a typical femtocell network is depicted in Fig. 5.4. We assume that mobile user u_l, connecting to the FAP f_i located at point A, is requesting file c_k and moving out of the corresponding coverage area and approaching to FAPs f_{i+1} and f_{i+2}. At the border of the current FAP f_i (point B in Fig. 5.4), when $RSSI_{i,l} < P_{th}$, handover becomes imminent. Then, the mobile user discovers two candidate neighboring FAPs for handover, for which $RSSI_{i+1,l}, RSSI_{i+2,l} > P_{th}$. The RSSI values of both FAPs f_{i+1} and f_{i+2} exceed the threshold due to their coverage overlap. The proposed

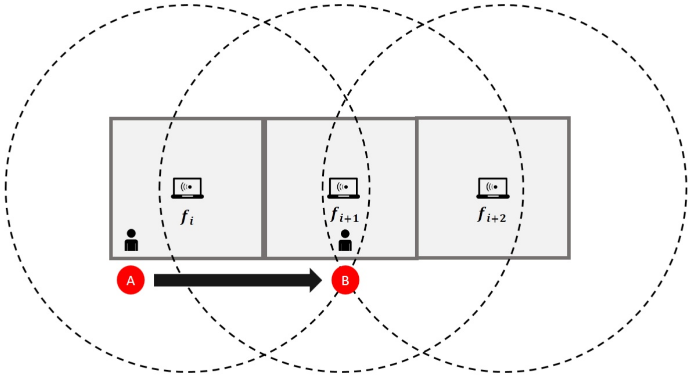


Figure 5.4: The main idea behind the proposed Mobility-Aware Femtocaching scheme based on Handover.

MAFH algorithm chooses FAP f_{i+2} instead of FAP f_{i+1} for downloading the remaining segments of video file c_k to avoid unnecessary handover. The benefit of long connection time between mobile users and FAPs is that it enables mobile users to download more encoded segments before next handover.

The main goals of our algorithm are to reduce the number of handover and increase the connection time between user u_l and FAP f_{i+2}. As can be seen in Fig. 5.4, during the movement of user u_l between points A and B, the RSSI from available femtocells, including f_i and f_{i+1}, are measured frequently to form the $NFL = \{f_i, f_{i+1}\}$. When user u_l is located at point B, handover is triggered and $CL = \{f_i, f_{i+1}, f_{i+2}\}$ is formed. Finally, the FAP with the minimum RSSI from the CL would be selected as the target. It is possible to select FAP f_i that user u_l has been already connected to because it shows minimum RSSI. This selection is perfectly legit since it shows user u_l is moving backward. In general, based on the presence of each FAP in NFL and/or CL and the corresponding value of RSSI, one of the following scenarios occurs to trigger handover:

Scenario 1: If the minimum RSSI is associated with a FAP, such as f_{i+2}, which is not in NFL, then the FAP is selected as the target for user u_l to connect to f_{i+2} and download remaining segments of file c_k.

Scenario 2: If the minimum RSSI is associated with a FAP, such as f_i, which is in NFL, the RSSI from FAP f_i in NFL is compared with its RSSI value in CL. If the RSSI in NFL is less than that in CL, it means user u_l is approaching to

f_i, therefore, the selected FAP f_i is the target and the communication between user u_l and FAP f_i is established.

Scenario 3: In contrast to scenario 2, if the RSSI from a specific FAP, such as f_i, in CL is less than of its RSSI in NFL, it indicates that u_l is moving away from the FAP; hence, that FAP is not an appropriate candidate for handover. In this case, FAP f_i will be removed from the CL and another FAP with the minimum RSSI is chosen.

To reduce the number of unnecessary handovers in our proposed MAFH algorithm, we classify handover decisions into two categories, named prompt handovers, and long handovers, as follows:

Prompt Handover: This would happen in a massive network, if the user connects to the nearest FAP with RSSI's highest value after triggering a handover. In this respect, if the user moves at high speed, there will be another early handover, as this user leaves FAP's current range of transmission.

Long Handover: In this case, the user connects to the further FAP with the lowest RSSI value, leading to expanding the connecting time between MU and the caching node.

In order to illustrate the reduction of handovers in our proposed scheme, we apply an optimization problem to the cost function, denoted by C_h, based on our algorithm, defined as follows [179]:

$$C_h(\mathbf{w}) = w_{ph}N_{ph} + w_{lh}N_{lh}, \tag{130}$$

where N_{ph} and N_{lh} represent the number of prompt and long handover, respectively, and $\mathbf{w} = [w_{ph}, w_{lh}]$ is the vector of weight factor that should be optimized. By considering the Taylor series expansion at optimum vector $\mathbf{w}^*$ to minimize the cost function with respect to $\mathbf{w}$, we have:

$$C_h(\mathbf{w}^*) \approx C_h(\mathbf{w}) + (\mathbf{w}^* - \mathbf{w})C_h^{'}(\mathbf{w}) + \frac{(\mathbf{w}^* - \mathbf{w})^2}{2}C_h^{''}(\mathbf{w}), \tag{131}$$

where $\mathbf{w}^*$ denotes the optimum vector, minimized C_h function, and $C_h^{'}$ and $C_h^{''}$ represent the first, and the second derivatives of C_h, respectively. In this case, the optimized value of $\mathbf{w}^*$ is obtained as follows:

$$\mathbf{w}^* = \mathbf{w} - \frac{C_h^{'}(\mathbf{w})}{C_h^{''}(\mathbf{w})}. \tag{132}$$

where w_{ph}^* and w_{lh}^* in the simulation result in α and β values.

Case 2 (Low Speed Users): In this case, user u_l that requests a content is either anchored at its place or moves slowly with the speed $v_l^t \leq v_{th}$ within the transmission range of a particular FAP. Therefore, user u_l does not experience any handover during a request. For this reason, low speed users have two options to access their contents; the first is to connect to a local FAP listed in NFL, and the second is to establish a connection with other nearby mobile devices that are anchored or move with a slow speed. Furthermore, they can access more segments of the required file if they connect to both FAPs and mobile devices. However, if a same segment of the requested content is available in both D2D and FAP caches, user u_l is served by the FAP to avoid consuming energy of another user by establishing D2D communication. To connect with FAPs, unlike previous case, low speed user u_l selects a FAP with the highest value of RSSI in CL. This allows user u_l to access the requested content from the target FAP f_{i+1} within the shortest possible time due to the short distance between u_l and f_{i+1}. For D2D communication, we need to consider the situations that MUs become close to each other or move farther away. Hence, we define the difference vector $\mathbf{\Delta}_l = [\Lambda_{l,1}, \ldots, \Lambda_{l,N_l}]$, where $\Lambda_{l,j}$ is obtained as follows:

$$\Lambda_{l,j} = RSSI_{l,j}(t+1) - RSSI_{l,j}(t), \qquad j = 1, \ldots, N_l, \tag{133}$$

where N_l indicates the number of candidate mobile devices in the vicinity of user u_l. Moreover, $RSSI_{l,j}(t)$ and $RSSI_{l,j}(t+1)$ are the strength of received signal from user u_j in two consecutive time slots [182]. A large value of $\Lambda_{l,j}$ shows MUs u_j and u_l become closer that will increase the connection time. Therefore, a D2D communication can be established between u_j and u_l. The pseudo-code of our proposed MAFH scheme is outlined in **Algorithm 3**.

Algorithm 3 Proposed MAFH Scheme

```
**Initialization:** Set $P_{th} = -67$ dBm and $\upsilon_{th} = 3600$ m/h.
**while** $u_l$ requests content $c_k$ **do**
    **Step 1 (Caching Nodes Identification):**
    Estimate the location of $u_l$ according to (17)-(20).
    **Step 2 (Speed Estimation):** Estimate $\upsilon_l^t$ according to
        $$d\boldsymbol{\upsilon}_l^t = -\begin{pmatrix} -\log\gamma_1 & \theta \\ -\theta & -\log\gamma_2 \end{pmatrix}(\boldsymbol{\upsilon}_l^t - \boldsymbol{\mu})dt + \mathbf{J}d\mathbf{B}_t,$$
    **for** $j = 1, \dots N$ **do**
        $$RSSI_{j,l} = P_j - PL_{j,l},$$
    **end for**
    **Step 3 (Type of Connection):**
    **if** $RSSI_{j,l} < P_{th}$ **then**
        **if** $\upsilon_l^t \geq \upsilon_{th}$ **then**
            Update NFL and CL
                $$NFL = \{f_j, j = 1, ..., n | RSSI_{j,l} > P_{th}\},$$
                $$CL = \{f_{j'}, j^{'} = 1, ..., n^{'} | RSSI_{j^{'},l} > P_{th}\},$$
            Select $f_m \in \{CL | RSSI_{CL}(f_m) = min(RSSI_{CL}(f_i)), \forall f_i \in CL\}$
            **if** $f_m \in NFL$ **then**
                **if** $RSSI_{NFL}(f_m) < RSSI_{CL}(f_m)$ **then**
                    Select $f_m$.
                **else**
                    Remove $f_m$ from the $CL$ and
                    choose FAP with the minimum RSSI.
                **end if**
            **else**
                Select $f_m$.
            **end if**
        **else**
            Select $f_m \in \{CL | RSSI_{CL}(f_m) = max(RSSI_{CL}(f_i)), \forall f_i \in CL\}$
            Connect to user $u_k$ with the highest value of
            $\Lambda_{l,k}$, in which $\Lambda_{l,k}$ is obtained as follows:
                $$\Lambda_{l,k} = RSSI_{l,k}(t+1) - RSSI_{l,k}(t), \qquad k = 1, \dots, N_l,$$
        **end if**
    **end if**
**end while**
**Output:** The type of communication between MU $u_l$ and caching nodes is determined.
```

5.1.3 Simulation Results

In this Subsection, we evaluate the effectiveness of the proposed algorithm in terms of the cache hit ratio, transferred byte volume, connecting time, content delivery time, and the number of handovers particularly for high speed users, as well as the energy consumption for low speed clients. Moreover, we assume the random walk model to present the users mobility pattern for the system model in Fig. 5.1. For the scenarios under simulation, we investigate how the connection type and mobility pattern affect the aforementioned performance metrics.

Simulation Setup: We consider an enterprise femtocell network consisting of 250 small stores which all of them are deployed with FAP and 1500 MUs distributed uniformly inside the area covered by the MBS with the radius $R =$ 500 m. The MBS located in the center of enterprise, based on the 4G LTE standard. It is assumed that the transmission range of each FAP is $r_f = 10$ m, therefore, all FAPs are overlapped with each other. There exist $M = 6000$ distinct contents in the network, with the size of $s_{c_k} = 300$ MB. In addition, the storage capacity of FAPs C_f is approximately 180 GB, which is equal to 10% the whole of network's content. Moreover, each file is fragmented into $T = 10$ input symbols, based on the Fountain Code, where the size of each segment is about $s_{c_{k,l}} = 30$ MB. It is supposed that user u_l at time t moves through the corridors and shops with a velocity υ_l^t based on the random walk model in Subsection 5.1.1 and request contents, while the probability of requesting content c_k follows the Zipf distribution with parameter $\lambda = 0.56$ [109]. To be more practical and regarding to the coverage area of FAPs, i.e., $r_f = 10$ m, we assume $\upsilon_{th} = 3600$ m/h, which is equivalent to $\upsilon_{th} = 1$ m/s. All users are capable of establishing D2D connection in the range of $r_u = 2$ m. The storage capacity of users involved in D2D communication is equal to the size of 100 contents, i.e., $C_u = 30$ GB. The general simulation parameters are summarized in Table I. In the proposed network model, we compare the performance of the following four methods with two existing caching schemes, namely, MAMAB [180] and MAFMC [181]:

Traditional Mobility-Aware Femtocaching Scheme with Forward Movement (MAFF): Without considering the velocity of users, the requesting client

Table 5.1: Simulation parameters settings

Frequency	2GHz
Macrocell TX Power	46 dBm
Femtocell TX Power	20 dBm
N	250
S	1500
M	6000
R	500 m
r_f	10 m
r_u	2 m
s_{c_k}	300 MB
$s_{c_{k,l}}$	30 MB
T	10
C_f	180 GB
C_u	30 GB
L_{ow}	20 dB
hm	4 dB
P_{th}	-67 dBm
υ_{th}	3600 m/h
λ	0.56
W_q	1.4 MHz
K	0.07
N_0	-174 dBm/Hz

will be connected to the nearest caching node, either FAP or another MU. In this case, clients just move forward through the corridors and shops.

Traditional Mobility-Aware Femtocaching Scheme with Random Walk Model (MAFR): In this case, we assume that the distance between requesting user and caching nodes is the only deterministic factor to prioritize caching nodes. However, by emphasizing on the implementation of a practical model, the movement pattern of clients is defined based on the random walk theory described in Subsection 5.1.1.

Mobility-Aware Femtocaching Scheme based on Handover with Forward Movement (MAFHF): This method was completely described in Subsection 5.1.2, with the difference that users just move straight forward through the corridors.

Mobility-Aware Femtocaching Scheme based on Handover with Random Walk Model (MAFH): This scheme is our proposed strategy, described in **Algorithm 3**, with focus on the random walk mobility pattern and handover phenomenon.

Taking the above considerations into account, we first investigate the accuracy rate of the TDoA scheme, as the efficient indoor localization method, employed in the proposed MAFH algorithm to determine the exact location of clients. Then, we illustrate the superiority of the proposed MAFH framework compared with the existing conventional schemes from the aspect of cache hit ratio, connecting time, the number of handovers, delivery time, and the energy consumption of clients in D2D communication.

TDoA Indoor Localization: The dense deployment of caching nodes in the enterprise network provides many opportunities for MUs to download the required content. Therefore, after requesting a content, it is essential to determine neighbor caching nodes, including FAPs and MUs, in the vicinity of requested user. In this scenario, we first evaluate the effectiveness of the TDoA scheme as an indoor localization method to estimate the exact location of clients in Fig. 5.5, in which clients are randomly distributed in $20 \times 20\ m^2$ area of the enterprise network. The accuracy of the TDoA estimation depends on some parameters, such as sampling rate, signal bandwidth, and the existence of Line of

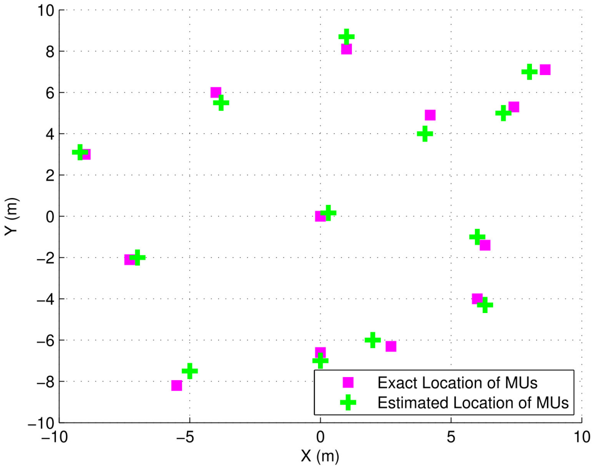


Figure 5.5: Location estimation results based on the TDoA indoor localization method.

Sight (LoS) signal between nodes. According to the results in Fig. 5.5, the estimated location of MUs are consistent and close to the real positions. Note that the positioning error is small, where the LoS signal between caching nodes and clients exists. To evaluate the accuracy of the TDoA scheme, the Root Mean Square Error (RMSE) is defined as $R_{RMSE} = \sqrt{\frac{1}{n}\sum_{l=1}^{S}(\hat{x}_l - x_l)^2 + (\hat{y}_l - y_l)^2}$, where (x_l, y_l), and $(\hat{x}_l, \hat{y}_l)$ are the true and the estimated coordinates of user u_l, respectively. By considering this fact, the RMSE of the TDoA method in our proposed network is about 0.47 m, with 4.7% estimation error.

Cache Hit Ratio: We evaluate the effect of different values of user's velocity υ_l^t on the performance of our proposed MAFH scheme and other three strategies mentioned above and compare the results with that of the MAMAB [180] and MAFMC [181] methods from the aspect of the cache hit ratio and transferred byte volume in Fig. 5.6. According to the results in Fig. 5.6a, since movement of low speed users is negligible, the mobility pattern and handover phenomena have not a remarkable effect on the cache hit ratio, therefore, the cache hit ratio for the values of $\upsilon_l^t < \upsilon_{th}$ follows a similar trend in all schemes. The most significant reason why the cache hit ratio decreases continuously for the values of $\upsilon_l^t \geq 1.7\upsilon_{th}$ in MAMAB, MAFR, and MAFF schemes is that for the constant value of contact duration $t_{l,q}^c$, the growth of user's speed leads to decreasing the time that users can be connected to the current caching node to download the segment of the required content. As a result, requesting user achieves less

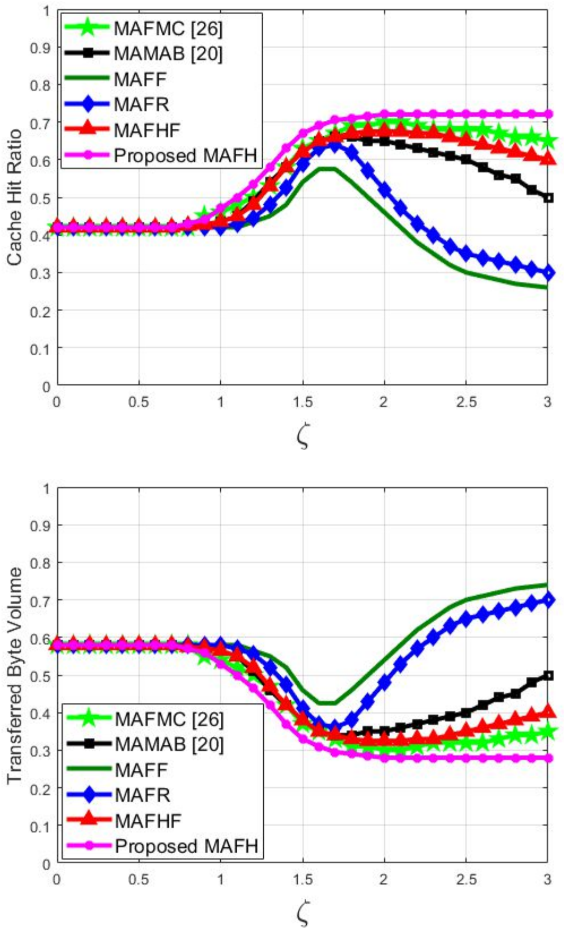


Figure 5.6: *a*) The cache hit ratio versus different values of $\zeta = \upsilon_l^t/\upsilon_{th}$, and *b*) the transferred byte volume versus different values of $\zeta = \upsilon_l^t/\upsilon_{th}$.

segments in one contact. By contrast, in the MAFHF and the proposed MAFH schemes based on handover, connection of high speed clients to the further FAPs leads to expanding the connecting time and downloading more segments. Furthermore, Fig. 5.6a illustrates the impact of the mobility pattern on the cache hit ratio. In the case that high speed users move according to the random walk theory in MAFH strategy and Markov chain in MAFMC algorithm, it is more fairly likely that users remain in the transmission range of the current caching node to increase the connecting time and attain more segments, while forward high speed users definitely leave the current caching node in a small time duration.

We also evaluate the performance of our proposed MAFH algorithm with other schemes from the aspect of transferred byte volume in Fig. 5.6b. Since the

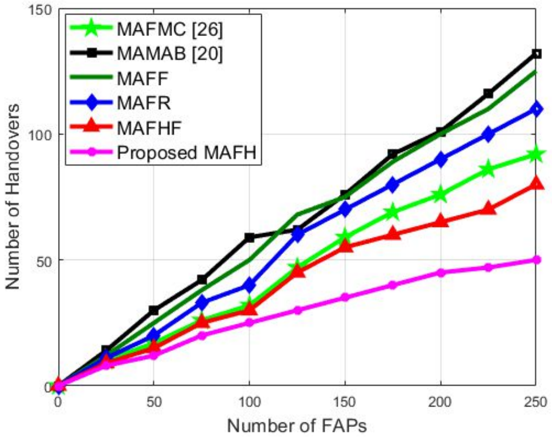


Figure 5.7: The number of handovers versus the number of deployed FAPs.

Table 5.2: The average length of the normalized connecting time of users in 100 requests.

MAMAB [20]	MAFMC [26]	MAFF	MAFR	MAFHF	MAFH
0.76	0.91	0.74	0.85	0.92	0.97

amount of data, which is downloaded from the MBS, is inversely proportional to the cache hit ratio, with the same reason as above, the proposed algorithm experiences better performance compared to other strategies.

Connecting Time: In this scenario, we evaluate the performance of the proposed MAFH algorithm with other schemes from the aspect of the average length of the normalized connecting time in 100 consecutive requests. As previously mentioned, connecting time is the time interval that mobile users can stay connected to the current caching node for downloading one segment of content, and that depends on the velocity of users. In the case of low speed users, they are capable of connecting to the current caching node as long as all available segments of the required content are completely downloaded. Therefore, there is no serious challenge for this type of clients. High speed clients, however, move and leave the transmission area of current FAP, and it is fairly likely that the possible connecting time is much less than the required time of contact for downloading a complete segment. According to the results in Table 5.2, due to the connection between high speed clients and further FAP, instead of near FAP, the proposed MAFH scheme has the best performance in terms of the

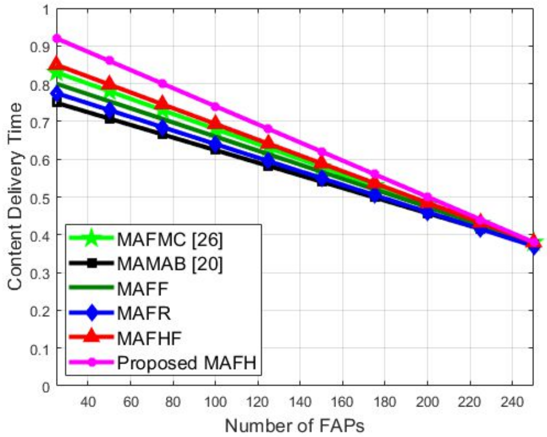


Figure 5.8: The content delivery time versus the number of deployed FAPs.

connecting time, which illustrates the extension of the average time that users connect to the current FAP. On the other hand, D2D connection in our proposed scheme is limited to low speed clients, staying in their current positions for a long time that leads to increasing the connecting time. Moreover, the random movement of clients in the proposed scheme increases the probability of staying in the transmission range of the current FAP to attain all available segments of the desired content.

Handover: In this part, we evaluate the number of handovers in the proposed MAFH scheme along with other traditional algorithms versus the number of deployed FAPs. It is obvious from Fig. 5.7 that increasing FAPs leads to growing the number of handovers. Nevertheless, the proposed MAFH algorithm postpones significantly the handover phenomenon by considering the random walk model for the mobility of clients and connecting to further FAP instead of near FAPs. Due to the random location and mobility of users in each run, the number of handovers in our proposed MAFH scheme is determined based on the optimum values of w_{ph}^{*} and w_{lh}^{*} across 1000 repeated simulations for a specific number of FAPs. It is evident that connecting to further FAP instead of near FAPs in the proposed MAFH strategy results in the reduction of the number of prompt handovers, followed by a remarkable decrease in the total number of handovers. Simulation results have shown almost 56% and 39% improvement in the number of handovers for the proposed MAFH algorithm when compared to the MAMAB and MAFMC methods, respectively.

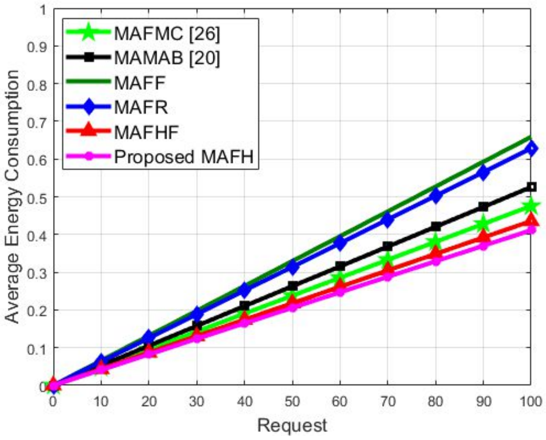


Figure 5.9: The energy consumption of clients during D2D communication.

Delivery Time: In this scenario, we demonstrate the effect of increasing the number of FAPs on the content delivery time in Fig. 5.8, which is expressed as the average time duration in order to download one segment of the required content. It is observed from Fig. 8 that increasing the number of deployed FAPs in all strategies leads to enhancing the content delivery time. In the proposed MAFH scheme, users connect to the further FAP, which leads to increasing the distance between clients and FAPs. Since the delay is directly related to the distance between transmitter and receiver nodes, the content delivery time raises slightly in handover femtocaching methods. However, as it can be seen from Fig. 5.8, the proposed algorithm displays almost the same behavior as the traditional mobility-aware femtocaching schemes, where the number of FAPs is more than 180 in the network.

Energy Consumption: Finally, we evaluate the performance of the proposed MAFH algorithm with other traditional schemes from the aspect of the energy consumption of clients during D2D communication in Fig. 5.9. The D2D communication reduces significantly the number of requests served by the MBS. However, this type of connection inherently decreases the life-time of users' battery. By considering this fact, it is essential to ignore the D2D communication, especially for high speed users that disconnect with other clients in a short time. As can be seen from Fig. 5.9, the proposed strategy has the best performance in terms of the energy consumption of clients. Numerical results have shown almost 21% and 13% improvement in the energy consumption of clients for the proposed MAFH algorithm when compared to the MAMAB [180]

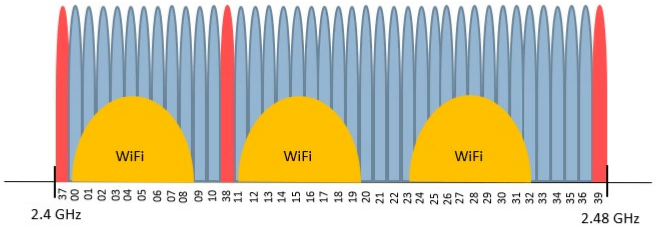


Figure 5.10: The 2.4 GHz frequency band shared by BLE and WiFi. Red channels are the advertisement and the blue ones are data channels used by BLE.

and MAFMC [181] methods, respectively.

5.2 Fusion Pre-Processing Techniques for BLE-based Indoor Localization

With expected widespread implementation of 5G networks and 5G Internet of Things (IoT), indoor localization is expected to become of even further importance. Although Global Positioning System (GPS) ensures efficient outdoor localization, generally speaking, indoor localization systems fail to provide the same level of efficiency. In this regard, there has been recent widespread attention to Angle of Arrival (AoA) with the application on Linear Antenna Array (LAA), as an efficient indoor localization method due to its potential in determining location with low estimation error. The AoA, however, suffers from several issues including being sensitive to multipath effects, noise, fluctuations of received signal, and frequency/phase shifts. To tackle these issues, this Section proposes a set of signal processing and information fusion methods by integration of Nonlinear Least Square (NLS) curve fitting, Kalman Filter (KF), and Gaussian Filter (GF) to boost the accuracy rate of estimated angle. The proposed fusion framework is evaluated based on a real Bluetooth Low Energy (BLE) dataset and results illustrate significant potentials in terms of improving overall BLE-based achievable accuracy in angle detection. In following, we, first introduce BLE wireless signal model, including BLE Transmitter, wireless channel, and BLE receiver.

5.2.1 BLE Wireless Signal Model

One of the noticeable challenges of wireless technologies for IoT applications is high power consumption. BLE as one of the prominent Wireless Personal Area Network (WPAN) standards has paved the way for emergence of IoT applications. BLE employs Gaussian Frequency Shift Keynig (GFSK) modulation, with the transmission power of -20 to $+20$ dBm. It operates in 2.4 to 2.48 GHz frequency range, which is the same spectrum as WiFi. It is designed to operate at a distance of 10 to 100 m, though typically at 10 m or less. It uses 40 channels, including 37 data channels and 3 advertisement channels with 2 MHz bandwidth. From the channel perspective, as it can be seen in Fig. 5.10, the spectrum of WiFi and BLE overlap with each other in several channels. Therefore, to obtain an accurate angle, it is essential to consider the interference of WiFi on BLE. By considering the distribution of interference and noise in different BLE channels, we can find the most reliable angles. Moreover, BLE's operation mode come with Frequency Hopping (FH) method, allowing the modules to jump on data channels to avoid interference within BLE channels.

There are two different modes for data transmission on BLEs, including broadcast mode and connected mode. Devices use the broadcast mode to advertise their presence, including some information about the device type, manufacturer, operating on advertisement channels. Once one of the advertised beacons has been received on the broadcast bands, another device (master) initiates a connection to this device (slave). The master and the slave devices agree on several connection parameters during the connection establishment process, one of which is the frequency hop distance, denoted by f_{hop}. In order to send each packet, master and slave devices hop through the 37 data channels by f_{hop} bands. Because the total number of channels is prime 37, the transmissions jump through all accessible bands before repeating a channel that was previously used [183]. Fig. 5.11 illustrates a typical structure of a BLE-based GFSK transceiver, including transmitter, wireless channel model, and receiver.

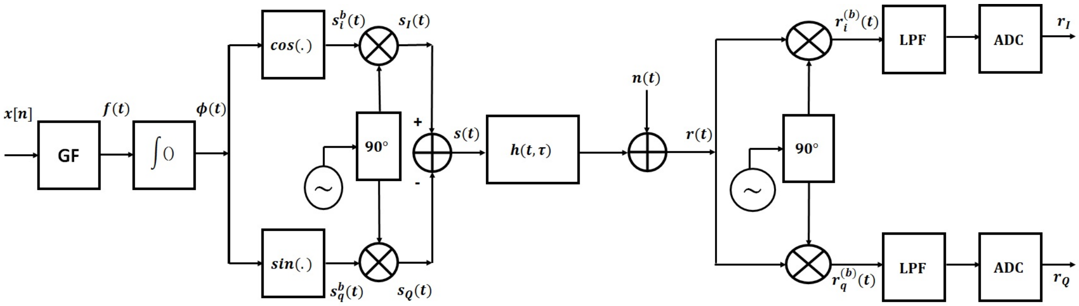


Figure 5.11: Structure of the transmitter, wireless channel model, and receiver in the BLE standard.

BLE Transmitter

The transmitted BLE signal is represented as

$$s(t) = \sqrt{\frac{2E}{T}} \cos\left(2\pi f_c t + \phi(t) + \phi_0\right), \tag{134}$$

where E, T, and f_c denote the energy and period of the transmitted symbol, and the carrier frequency, respectively. Since the GFSK modulation only affects the phase of the signal, the amplitude of $s(t)$, denoted by $\sqrt{2E/T}$ would be constant. Term ϕ_0 represents the initial phase of the incident signal. Moreover, the phase deviation $\phi(t)$ is obtained as

$$\phi(t) = \frac{\pi h}{T} \int_{-\infty}^{t} \sum_{n=-\infty}^{+\infty} x[n] p(\tau - nT) d\tau, \tag{135}$$

where h denotes the modulation index, defined as the ratio of the peak to peak frequency deviation to bit rates. Note that the bandwidth of the signal depends on the value of h, where the greater the value results in the wider bandwidth. It is commonly assumed in BLE standard that h is between 0.45 to 0.55. Additionally, $x[n]$ is the original baseband pulse sequence, which could be $+1$ and -1. The Gaussian Filter (GF) $p[k]$ is implemented in discrete-time domain with a sample period of T_s on the baseband pulse sequence $x[n]$, where the symbol period is T. In this case, $p[k] = p(T_s k)$ is obtained as follows

$$p(t) = \frac{hT_s}{2\sigma\sqrt{2\pi}} e^{\frac{-t^2}{2\sigma^2}} \circledast \text{rect}(T, t), \tag{136}$$

where $\circledast$ is the convolution operator. Term σ is $0.13T/(BT)$, where B is the 3 dB bandwidth of the GF (in BLE standard $BT = 0.5$). Additionally, $\text{rect}(T,t)$ is obtained as follows

$$\text{rect}(T,t) = \begin{cases} \dfrac{1}{T}, & \dfrac{-T}{2} \leqslant t \leqslant \dfrac{T}{2} \\ 0, & \text{otherwise} \end{cases} . \tag{137}$$

Note that $s(t) = Re\{s^b(t)e^{j2\pi f_c t}\}$, where $s^b(t)$ represents the baseband version of the transmitted signal, is obtained as

$$\begin{aligned} s^b(t) &= s_i^b(t) + js_q^b(t) \\ &= \sqrt{\frac{2E}{T}}\Big\{\cos(\phi(t)+\phi_0) + j\sin(\phi(t)+\phi_0)\Big\}. \end{aligned} \tag{138}$$

Then, we have $s(t) = Re\{s^b(t)e^{j2\pi f_c t}\} = s_i^b(t)cos(2\pi f_c t) - s_q^b(t)sin(2\pi f_c t) = s_I(t) - s_Q(t)$, where

$$s_I(t) = \sqrt{\frac{2E}{T}}\{cos(2\pi f_c t)cos(\phi(t)+\phi_0)\} = \tag{139}$$
$$\sqrt{\frac{E}{2T}}\{cos(2\pi f_c t + \phi(t) + \phi_0) + cos(2\pi f_c t - \phi(t) - \phi_0)\},$$

and

$$s_Q(t) = \sqrt{\frac{2E}{T}}\{sin(2\pi f_c t)sin(\phi(t)+\phi_0)\} = \tag{140}$$
$$\sqrt{\frac{E}{2T}}\{cos(2\pi f_c t - \phi(t) - \phi_0) - cos(2\pi f_c t + \phi(t) + \phi_0)\},$$

Wireless Channel

The transmitted wireless signal propagating in indoor environment gets reflected, refracted, diffracted, and scattered due to encountering different objects. Consequently, wireless signals become weaker in relation to the distance they travel, known as the path loss effect. Moreover, receiver collects a variety of diverse phase delayed and power attenuated versions of the transmitted

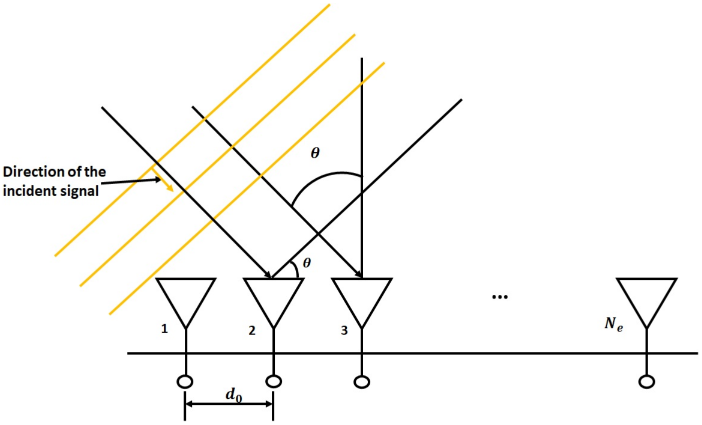


Figure 5.12: Structure of a typical Linear Antenna Array (LAA).

signal. Complex Finite Impulse Response (FIR) filter, denoted by $h(t,\tau)$ is implemented to model and simulate the wireless multipath fading channel. In this regard, the baseband channel impulse response, denoted by $h_b(t,\tau)$ is calculated as follows

$$h_b(t,\tau) = \sum_{k=1}^{N(t)} \rho_k(t,\tau) e^{-j\theta_k(t)} \delta(t - \tau_k(t)), \tag{141}$$

where $N(t)$ is the number of resolvable multipath components and $\rho_k(t,\tau)$ denotes the attenuation of the k^{th} path. Moreover, $\tau_k(t)$ and $\theta_k(t) = 2\pi f_c \tau_k(t)$ represent the delay of the k^{th} path and its associated phase shift, respectively.

BLE Receiver

The data being transmitted is modulated on the radio frequency carrier, so that at the receiver it can be extracted accurately. As can be seen from Fig. 5.12, there are N_e number of elements, aligned with the same distance d_0 in a LAA as the receiver. Once an MSA send data to BLE beacons, the incident signal is received by different elements in the LAA. As stated previously, there are many different interrupting factors that have a destructive impact on the transmitted signal, including CFO, ISI, noise, and phase shifting. CFO occurs when the receiver's local down-conversion oscillator, denoted by f_r, fails to synchronise with the carrier frequency in the transmitted end f_c. Occurrence of CFO is expected as the incident signal in the BLE standard transmits between 2.4 and 2.48 GHz over one of the 37 data channels while the receiver oscillator is tuned to a pre-defined frequency. As a result, the angle of the oscillator signal shifts around the expected instantaneous phase. In addition, noise strongly affects the efficiency of the wireless communication systems, which is modelled as Additive White Gaussian Noise (AWGN), $n \sim \mathcal{N}(0, \sigma^2)$. By considering all these effects, the baseband received signal, denoted by $r_b(t)$ is obtained as

$$r_b(t) = \big(s_b(t) * h_b(t, \tau)\big)\frac{1}{2}e^{j2\pi(f_c - f_r)t + \phi_c(t)} + n(t), \tag{142}$$

where $\phi_c(t)$ denotes the phase shift. In the frequency flat channel, the delay spread is small compared to the symbol duration. Therefore, it can be assumed that $s(t - \tau_k(t)) = s(t)$. To characterize the channel, we define $\alpha(t)$ based on Eqs. (141) and (142) as follows

$$\alpha(t) = \sum_{k=1}^{N(t)} \rho_k(t, \tau) e^{-j\theta_k(t) + j2\pi f_c t + \phi_c(t)} = x(t) + jy(t). \tag{143}$$

The received signal, therefore, can be expressed as follows

$$r(t) = \alpha(t)s(t) + n(t). \tag{144}$$

If the carrier frequency of the transmitter and the receiver is exactly the same, the value of the in-phase and the quadrature part of the received signal should be a constant. However, because of the oscillator manufacturing process there will be a frequency gap. Therefore, the real $r_i^{(b)}(t)$ and imaginary $r_q^{(b)}(t)$ components of the baseband received signal are expressed as

$$\begin{aligned} r_i^{(b)}(t) &= r(t)\cos(2\pi f_r t) = \alpha(t)s(t)\cos(2\pi f_r t) + n_i(t) \\ &= \sqrt{\frac{E}{2T}}\alpha(t)\big\{\cos(2\pi(f_c - f_r)t + \phi(t) + \phi_0) + \\ &\qquad \cos(2\pi(f_c + f_r)t + \phi(t) + \phi_0)\big\} + n_i(t), \end{aligned} \tag{145}$$

$$\begin{aligned} r_q^{(b)}(t) &= r_q(t)\sin(2\pi f_r t) = \alpha(t)s(t)\sin(2\pi f_r t) + n_i(t) \\ &= \sqrt{\frac{E}{2T}}\alpha(t)\big\{\sin(2\pi(f_c + f_r)t + \phi(t) + \phi_0)) - \\ &\qquad \sin(2\pi(f_c - f_r)t + \phi(t) + \phi_0)\big\} + n_q(t). \end{aligned} \tag{146}$$

After passing through a Low-Pass (LP) filter and Analog to Digital Converter (ADC), the discrete received signals by different elements are expressed as

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,N_e} \\ \vdots & \ddots & \vdots \\ r_{N,1} & \cdots & r_{N,N_e} \end{bmatrix}, \tag{147}$$

where $r_{l,k} = r_{I_{l,k}} + jr_{Q_{l,k}}$. Terms N and N_e represent the number of samples per signal and the number of elements in LAA, respectively. Additionally, k^{th} column of $\mathbf{R}$ indicates samples received by element e_k. To calculate the phase difference between different elements, the magnitude/phase representation is utilized. In this regard, $\mathbf{R}$ is converted into $\mathbf{X}$, where $x_{l,k} = r_{A_{l,k}} e^{jr_{\phi_{l,k}}}$ with $r_{\phi_{l,k}}$ and $r_{A_{l,k}}$ are computed as

$$r_{\phi_{l,k}} = \arctan\left(\frac{r_{Q_{l,k}}}{r_{I_{l,k}}}\right), \tag{148}$$

$$r_{A_{l,k}} = (r^2_{I_{l,k}} + r^2_{Q_{l,k}})^{\frac{1}{2}}. \tag{149}$$

Consequently, samples gathered by element e_i, denoted by $\mathbf{x}_i = [x_{1,i}, \ldots, x_{N,i)}]^T$, is multiplied by the complex conjugate of samples from element e_j, denoted by $\mathbf{x}_j$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the i^{th} and j^{th} columns of matrix $\mathbf{X}$. The resulting phase difference is given by

$$\boldsymbol{\Delta\Phi}^{(i,j)} = \mathbf{x}_i \times \mathbf{x}_j^* \tag{150}$$

where $\Delta\Phi^{(i,j)}_{l,1} = r_{A_{l,i}} r_{A_{l,j}} e^{j(r_{\phi_{l,i}} - r_{\phi_{l,j}})}$. Therefore, the angle of incident signal can be calculated as

$$\boldsymbol{\theta}^{(i,j)} = \arcsin \frac{\lambda \boldsymbol{\Delta\Phi}^{(i,j)}}{2\pi d_{i,j}}, \tag{151}$$

where $\lambda = c/f_c$ and f_c denote the wave length and the carrier frequency of signal, respectively. Moreover, c is the speed of light, about $3 \times 10^8 m/s$, and $d_{i,j} = |j - i| d_0$.

To calculate the location of MSAs, the AoA scheme uses triangulation theory.

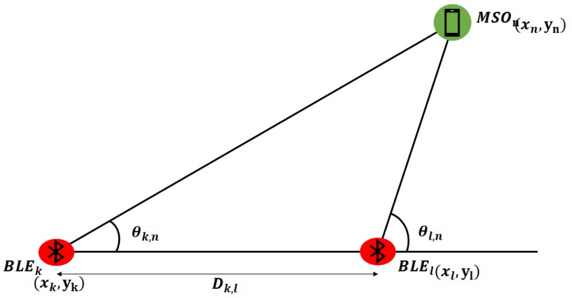


Figure 5.13: Location estimation based on the AoA method.

Use of at least two BLE beacons with known positions in this approach contributes to continuous tracking. It is assumed that the incident signal from MSA_n with the coordinate of (x_n, y_n) is received by B BLE beacons with the coordinate of (x_b, y_b), for $(1 \leqslant b \leqslant B)$. As it can be seen from Fig. 5.13, $\theta_{b,n}$, which is obtained according to Eq. (151), indicates the angle between x-axis and the line between MSA and BLE beacon. Location of the MSA is estimated as

$$x_n = \frac{D_{k,l} \tan \theta_{l,n}}{\tan \theta_{l,n} - \tan \theta_{k,n}}, \tag{152}$$

$$y_n = \frac{D_{k,l} \tan \theta_{k,n} \tan \theta_{l,n}}{\tan \theta_{l,n} - \tan \theta_{k,n}}, \tag{153}$$

where $D_{k,l}$ is the distance between the l^{th} and k^{th} BLE beacons. This completes presentation of the BLE wireless signal model. Next, we present our proposed BLE processing fusion framework.

5.2.2 Proposed Wireless Signal Processing/Fusion Framework

Our system model consists of user devices that are required to be found (localized), and known location BLEs embedded with LAA. Once a user device send data to the BLE, the incident signal is received by different elements in LAA. It is assumed that incident signal consists of different packets, where each packet has a length of N_p samples, and N samples are received by one element before switching to the next element. After sending packet p_i through the data channel with the frequency f_j, the next packet p_{i+1} would be sent through $f_{j+1} = f_j + f_{hop}$. The wireless signal received at the BLE beacon, however, is the distorted version of the transmitted signal due to the deregulation of transmitter and receiver and the multipath and pathloss effects. Obtaining phase difference based on the raw data, therefore, leads to a noticeable error in angle estimation. For this reason, the first step towards angle detection is data processing, including phase compensation.

Noise Robustness

Owing to the existence of reflective obstacles, walls and movement of people, indoor environments suffer from multipath, shadowing, and pathloss effects. Therefore, BLEs receive a sum of different transmitted signal versions, where the interference of a large number of signals can cause either signal amplification or attenuation. For this reason, we eliminate this effect by applying Nonlinear Least Squares (NLS) curve fitting on the raw data. Given a function $f(t)$ of a variable t, a received packet p_i with N_p samples, including I/Q samples can be represented by

$$\begin{aligned} f_I(t; A_I, f_c, \phi_{I,0}) &= A_I \sin(2\pi f_c t + \phi_{I,0}) \quad (154) \\ \text{and } f_Q(t; A_Q, f_c, \phi_{Q,0}) &= A_Q \sin(2\pi f_c t + \phi_{Q,0}). \quad (155) \end{aligned}$$

Our goal is to solve Eqs. (154) and (155) to obtain the best value of $\gamma_I \equiv (A_I, f_c, \phi_{I,0})$ and $\gamma_Q \equiv (A_Q, f_c, \phi_{Q,0})$ to diminish the effect of noise on the received signal. Toward this goal, first we pick an initial guess for γ_I and γ_Q, and define

$$\begin{aligned} d\beta_I &\triangleq r_I(t) - f_I(t; A_I, f_c, \phi_{I,0}), \quad (156) \\ \text{and } d\beta_Q &\triangleq r_Q(t) - f_Q(t; A_Q, f_c, \phi_{Q,0}). \quad (157) \end{aligned}$$

To obtain a linearized estimate for the changes of γ_I and γ_Q, it is essential to achieve $d\beta_I = 0$ and $d\beta_Q = 0$. In this regard, we have $d\boldsymbol{\beta} = \mathbf{A} d\boldsymbol{\gamma}$, where $\mathbf{A}$

denotes a (2×3) matrix is defined as

$$\mathbf{A} = \left[\begin{array}{ccc} \frac{\partial f_I}{\partial A_I}|_{\gamma_I} & \frac{\partial f_I}{\partial f_c}|_{\gamma_I} & \frac{\partial f_I}{\partial \phi_{I,0}}|_{\gamma_I} \\ \frac{\partial f_Q}{\partial A_Q}|_{\gamma_Q} & \frac{\partial f_Q}{\partial f_c}|_{\gamma_Q} & \frac{\partial f_Q}{\partial \phi_{Q,0}}|_{\gamma_Q} \end{array} \right]. \tag{158}$$

By applying the transpose of $\mathbf{A}$ to both sides, we have

$$\mathbf{A}^T d\boldsymbol{\beta} = (\mathbf{A}^T \mathbf{A}) d\boldsymbol{\gamma}. \tag{159}$$

Eq. (159) can be solved by applying standard matrix techniques such as Gaussian elimination. The best value of $\boldsymbol{\gamma}$ is obtained by using this method iteratively until $d\boldsymbol{\gamma}$ is smaller than a pre-defined threshold.

Eliminating Switching Error

As mentioned previously, we calculate the angle of incident signal based on the phase difference obtained from different elements. However, due to the phase shift of oscillator in both transmitter and receiver sides and also switching of elements, the value of $\boldsymbol{\Delta\Phi}^{(i,j)}$ for different i, j is not the same, leading to different values of $\boldsymbol{\theta}^{(i,j)}$. To solve this problem, we apply Kalman Filter (KF) on the phase difference obtained in different snapshots. To formulate the KF, an intermediate state vector $\boldsymbol{\Delta\hat{\Phi}}^{(i,j)}$ is defined, modeling the smoothed phase difference as the state-model given by

$$\boldsymbol{\Delta\hat{\Phi}}^{(i,j)}(k) = \boldsymbol{\Delta\hat{\Phi}}^{(i,j)}(k-1) + \boldsymbol{\rho}(k). \tag{160}$$

On the other hand, the observation model is defined by $\boldsymbol{\Delta\Phi}^{(i,j)} = \boldsymbol{\Delta\hat{\Phi}}^{(i,j)}(k) + \boldsymbol{\tau}(k)$, where the measured phase difference $\boldsymbol{\Delta\Phi}^{(i,j)}$ is used as the input observation to the KF. Terms $\boldsymbol{\rho}(k)$ and $\boldsymbol{\tau}(k)$ are Gaussian zero-mean variables used in the smoothing model with their second-order statistics ($\boldsymbol{Q}$ and $\boldsymbol{R}$) being obtained through an initial calibration phase.

Elimination of Frequency Error

There are 37 BLE channels to transmit data packets. However, it is evident that the angle obtained from different channels are different. One of the most important reason for that would be the interference of WiFi channels with BLEs. Therefore, it is essential to consider the impact of BLE's channel on the accurate angle. Toward this goal, after the calculation of phase difference based on Eq. (160), and calculating $\boldsymbol{\theta}^{(i,j)}$ in Eq. (139), the compensation vector $\boldsymbol{\chi}$ is defined to add to the estimated $\boldsymbol{\theta}^{(i,j)}$ to compensate the impact of BLE's channel. In this regard, we divide $-90^\circ \leq \theta \leq 90^\circ$ into 36 intervals, where $\theta_k = 5k - 90^\circ$, $k \in \{0, \ldots, 36\}$. For $-90^\circ \leq \theta \leq 90^\circ$ we obtain $\theta^{estimate}$ in all 37 channels in l iterations, and calculate the Mean Square Error (MSE) as

$$\omega_{k,ch} = \frac{1}{l} \sum_{q=1}^{l} (\theta_k - \theta_{ch,q}^{estimate})^2, \tag{161}$$

where $ch = 0, \ldots, 36$, and $k = 0, \ldots, 36$. In this case, we have a $(N_a \times N_{ch})$ coefficient matrix, denoted by $\boldsymbol{\omega}$, where N_a and N_{ch} denote the number of tested angles and the number of data channels in BLE technology, respectively. In this case, l^{th} column of matrix $\boldsymbol{\omega}$ denotes the MSE of angle θ_k in different channels. To obtain χ_{ch} associated with the compensation vector $\boldsymbol{\chi}$, we apply a Gaussian Filter (GF), denoted by $\mathcal{N}(\mu_{ch}, \sigma_{ch})$ on l^{th} column of matrix $\boldsymbol{\omega}$. Then, we obtain the final phase difference θ_{ch}^{final} as follows

$$\theta_{ch}^{final} = \theta_{ch}^{estimate} + \chi_{ch}. \tag{162}$$

5.2.3 Experimental Results

The proposed wireless signal processing methods are applied on a real dataset consisting of I/Q samples collected by *SimpleLink Angle of Arrival Booster Pack*. As shown in Fig. 5.14, this package includes RTLS master, RTLS slave, and RTLS Passive consisting of two linear antenna arrays with three elements. Due to availability of two linear antenna arrays, RTLS Passive can support

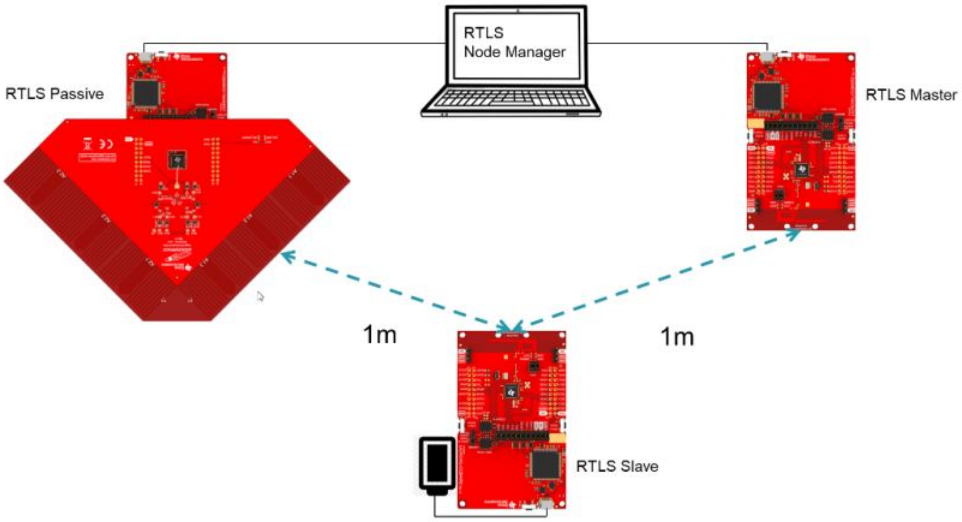
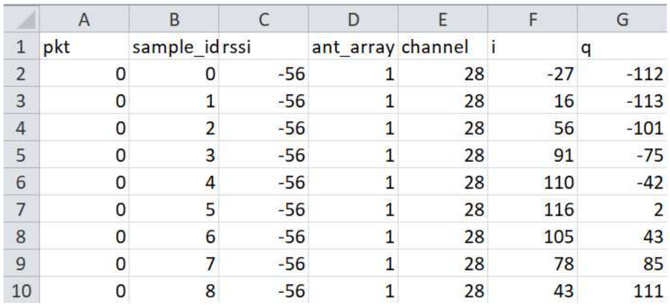


Figure 5.14: The structure of RTLS Device, including RTLS Passive, Slave, and Master devices [210].

an area with $-180° \leq \theta \leq 180°$. Without loss of generality, we analyze the data obtained from one antenna array to support an area within $-90° \leq \theta \leq 90°$. Fig. 5.15 illustrates all data, we have from a transmission signal. The carrier frequency and space distance between each elements d_0 are 2.4 GHz and 3.5 cm, respectively. The sample frequency is 250 kHz. At the beginning of the connection, the RTLS master shares some connection parameters with the computer. The RTLS slave is a device that we want to find its location, and RTLS passive receives packets from the slave device in terms of I/Q samples. Each data packet consists of 511 I/Q samples, where each $N = 16$ samples received by one element, before switching to another one. We eliminate the last 31 samples in each packet, since there is no switching in transmission of these samples. It is assumed that the first packet is sent randomly on one of the BLE's channel, however, other packets are sent on the channels, selected based on the FH method with $f_{hop} = 6$. After transmitting a packet completely, i.e., $N = 511$ samples, the received I/Q samples are fitted to sinusoidal signals based on the NLS curve fitting to eliminate the effect of noise on the data. Fig. 5.16 illustrates the real Inphase data and curve fitted signal.

To obtain the angle of arrival, we need to calculate the phase difference between signals received by different elements. In our case of study, the number of elements in an array antenna in the LAA is equal to 3. Therefore, we have 3 phase differences in each snapshot, including $\mathbf{\Delta\Phi}^{(1,2)}$, $\mathbf{\Delta\Phi}^{(2,3)}$, and $\mathbf{\Delta\Phi}^{(1,3)}$,



	A	B	C	D	E	F	G
1	pkt	sample_id	rssi	ant_array	channel	i	q
2	0	0	-56	1	28	-27	-112
3	0	1	-56	1	28	16	-113
4	0	2	-56	1	28	56	-101
5	0	3	-56	1	28	91	-75
6	0	4	-56	1	28	110	-42
7	0	5	-56	1	28	116	2
8	0	6	-56	1	28	105	43
9	0	7	-56	1	28	78	85
10	0	8	-56	1	28	43	111

Figure 5.15: The raw received data sent from slave nodes received by LAA in passive RTLS.

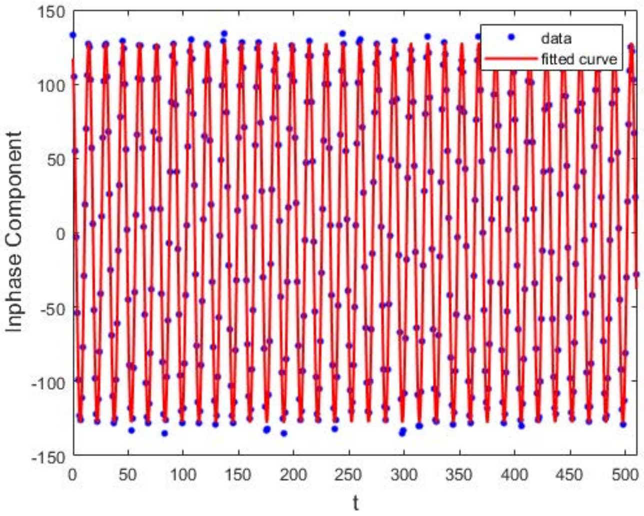


Figure 5.16: Curve fitting for Inphase samples of a packet with the length of 511.

where $\mathbf{\Delta\Phi}^{(1,2)} = \mathbf{\Delta\Phi}^{(2,3)} = \dfrac{1}{2}\mathbf{\Delta\Phi}^{(1,3)}$, since $d_{1,3} = 2d_{1,2} = 2d_{2,3}$.

As it can be seen from Fig. 5.17, there are specific phase differences between signals received by different elements. However, the value of phase difference is not equal for all samples in a particular packet. In addition, the value of $\mathbf{\Delta\Phi}^{(i,j)}$ for different i, j is not the same, leading to a remarkable error in the angle detection. This problem is tackled in our work by applying the KF on the phase difference obtained from different elements. To illustrate the performance of our processing method, we select 3 snapshots among $m = 10$ snapshots collected during one packet, including the first, the middle and the last one, to investigate the phase difference between different elements. According to the results shown in Tables 5.3 and 5.4, by applying NLS curve fitting and KF on raw data, the phase difference obtained from different samples and different

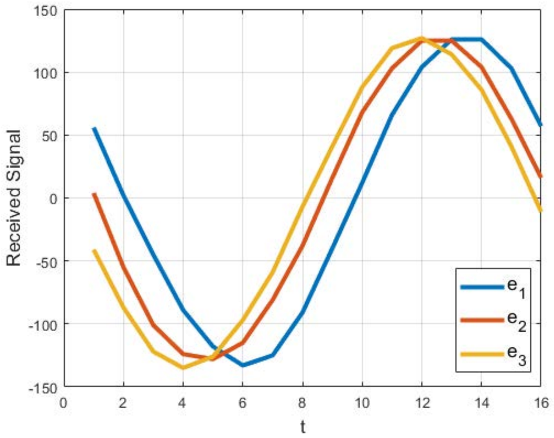


Figure 5.17: Phase difference between signals received by different elements in LAA. elements are the same, resulting in a low error in angle detection.

Table 5.3: Phase differences obtained from different elements before applying KF, where $\theta_{real} = -4$ degree, and the real value of phase differences should be equal to $\mathbf{\Delta\Phi}^{(1,2)} = -8$, $\mathbf{\Delta\Phi}^{(2,3)} = -8$, and $\mathbf{\Delta\Phi}^{(1,3)} = -16$ degree.

Type	1^{st} snapshot	5^{th} snapshot	10^{th} snapshot
$\mathbf{\Delta\Phi}^{(1,2)}$	-3.9423	-15.3012	-10.2659
$\mathbf{\Delta\Phi}^{(2,3)}$	-11.3190	-12.9405	3.5454
$\mathbf{\Delta\Phi}^{(1,3)}$	-15.2613	-28.2418	-6.7205

Considering the fact that there are 37 data channels for transmitting signal, each packet is being sent on different frequency, in which $f_c = 2.4 \times 10^9 + (\kappa + 2) \times 2 \times 10^6$ for $(0 \leq \kappa \leq 10)$, and $f_c = 2.4 \times 10^9 + (2\kappa + 3) \times 10^6$ for $(11 \leq \kappa \leq 36)$. Due to the same frequency bandwidth between WiFi and BLE technologies, BLE technology suffers from interference with WiFi. As a result, the value of θ needs to be compensated after the AoA is measured due to variability in results through frequency. Therefore, a constant offset $\boldsymbol{\chi}$, which is calculated based on the GF is applied on different frequencies to boost performance of the BLE-based AoA estimation. Table 5.5 shows the compensation values used for each frequency.

Fig. 5.18 compares the angle estimation error of our proposed processing method with such a case where there is no processing techniques on raw data. Accordingly, our proposed framework significantly reduces the AoA error and displays

Table 5.4: Phase differences obtained from different elements after applying KF, where $\theta_{real} = -4$ degree, and the real value of phase differences should be equal to $\mathbf{\Delta\Phi}^{(1,2)} = -8$, $\mathbf{\Delta\Phi}^{(2,3)} = -8$, and $\mathbf{\Delta\Phi}^{(1,3)} = -16$ degree.

Type	1st snapshot	5th snapshot	10th snapshot
$\mathbf{\Delta\Phi}^{(1,2)}$	-8.1851	-8.1813	-8.2438
$\mathbf{\Delta\Phi}^{(2,3)}$	-8.1704	-8.1824	-8.2539
$\mathbf{\Delta\Phi}^{(1,3)}$	-16.3555	-16.3909	-16.5578

Table 5.5: Frequency compensation $\boldsymbol{\chi}$ over all BLE channels.

Channel	37	0	1	2	3
χ_{ch}	0	-2	-4	7	15
Channel	4	5	6	7	8
χ_{ch}	18	14	8	6	5
Channel	9	10	38	11	12
χ_{ch}	1	1	0	3	5
Channel	13	14	15	16	17
χ_{ch}	8	9	10	13	11
Channel	18	19	20	21	22
χ_{ch}	-5	-3	-2	1	0
Channel	23	24	25	26	27
χ_{ch}	3	5	8	12	15
Channel	28	29	30	31	32
χ_{ch}	15	-7	5	8	4
Channel	33	34	35	36	39
χ_{ch}	2	3	1	-4	0

errors of less than 10° on most frequencies from −60° to 60°. It is worth mentioning that when the signal propagation direction is near parallel to the antenna array, the phase difference are almost random. Therefore, there exists a considerable error from −60° to −90° and 60° to 90° for most frequencies, due to the hardware setup of our device.

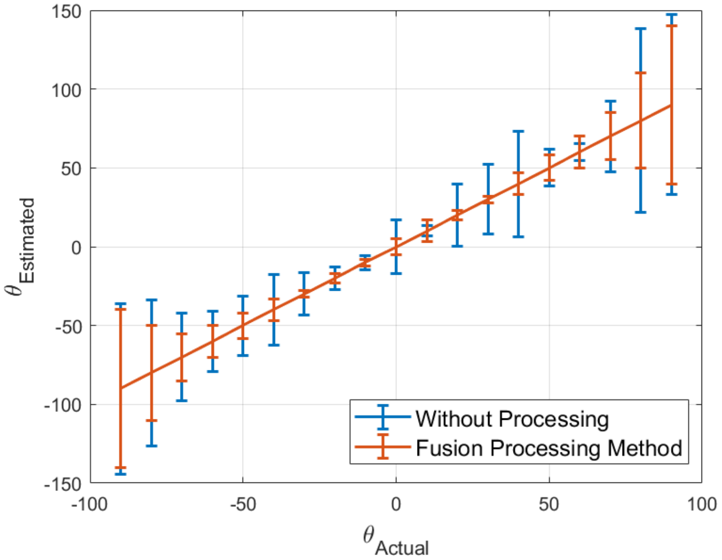


Figure 5.18: Compensated AoA after applying NLS curve fitting, KF, and GF.

5.3 Fusion Pre-Processing in Presence of Rayleigh Fading Channel

In this Section, we seek to investigate the main challenges of indoor environment, including Carrier Frequency Offset (CFO), Inter-Symbol Interference (ISI), multipath effect, noise, and phase shifting faced by the AoA. By considering the fact that there are a large number of obstacles in the indoor environment, it is fairly unlikely that there would be a Line of Sight (LoS) propagation between a mobile smart agent and BLE beacons. For the first time in the literature, to the best of our knowledge, we simulate the wireless channel of the BLE-based indoor environment by Rayleigh fading channel accompanied by Gaussian Frequency Shift Keying (GFSK) modulation. The BLE interface employs GFSK modulation technique, which is a spectrally efficient modulation form for digital communication. GFSK modulator is almost the same with FSK modulator, in which different carrier frequencies are assigned depending on the transmitted bit, with a difference that a Gaussian Filter (GF) is applied on the square-wave signal before modulation. Despite all the benefits that come with GFSK, including sideband power reduction resulting in low interference with neighboring BLEs' channels, the GFSK modulation suffers from the ISI issue. Towards this goal, we employ fusion processing techniques on raw data prior to utilization.

The proposed processing techniques consists of the following three phases:

- *NLS Curve Fitting*, which is applied on the received signal to eliminate the impact of noise.
- *Zero Forcing (ZF) Equalizer*, which is introduced to compensate the effect of Rayleigh channel fading.
- *Fast Fourier Transform (FFT)-based Phase Measurement*, which is applied on the received signal, before measuring the phase difference, as the transmitter frequency is not exactly known.

The effectiveness of our proposed processing method is evaluated via an experimental testbed, where In-phase and Quadrature (I/Q) samples, modulated by GFSK, are collected by BLE beacons. Additionally, there are Vicon cameras in our testbed to provide ground truth. With the knowledge of the precise location of MSAs during their movement, we consider the effect of noise and Rayleigh fading channel on the transmitted signal. By employing our proposed fusion processing techniques, we track MSAs with high accuracy.

To obtain the AoA based on the phase difference, it is essential to eliminate the impact of noise and fading channel on the received signal. Towards this goal, we propose a fusion framework consisting of the following three phases:

5.3.1 Noise Robustness

We apply NLS curve fitting on the raw received data to eliminate the effect of noice, which is described in the previous Subsection.

5.3.2 Zero Forcing Equalization

Equalizer is used to minimise the ISI and counteract the distortion caused by a signal transmitted through the fading channel. Since it is assumed that the multipath fading channel in this work is FIR, i.e., different delayed-versions of a same signal are received, we can use the ZF equalizer as the Infinite Impulse Response (IIR) filter. In this case, the effect of the estimated channel, denoted by $\alpha(t)$, on the received signal is compensated by the transfer function of the equalizer, i.e.,

$$H_{eq}(jw) = \frac{1}{\mid H(jw) \mid} e^{-j\angle H(jw)}, \qquad \mid w \mid \leqslant W_s \tag{163}$$

where $H(jw)$ is the transfer function of the channel, and W_s is the signal bandwidth [184].

5.3.3 Phase Compensation

After eliminating the effects of noise and fading channel on the received signal, we need to calculate the phase difference between two signals received by different elements. In this regard, first the DC offset of the compensated signals are removed. Then, by applying FFT on each signal and taking the complex point in each FFT, the phase difference between two points is calculated according to Eq. (150).

5.3.4 Simulation Results

To evaluate the proposed fusion framework, we used an experimental testbed consisting of four BLE beacons, located at the corner of a (4.5×4.5) m^2 rectangular indoor environment, and four Vicon cameras to provide ground truth. As a proof of concept experiment, we focus on tracking a single Mobile Smart Agent (MSA) moving in our environment based on the phase difference between I/Q samples collected by LAA. Extensions to the scenario with several MSAs moving in the environment possibly in presence of adversarial agents and/or cyber attacks (such as Denial of Service (DOS) attacks to disrupt cooperation within the MSAs) is the focus of our ongoing research.

There are three elements in each LAA, where the distance between two elements is $\lambda/2$. With the knowledge of precise location of the MSA during its movement, I/Q samples are generated according to the GFSK modulation. We evaluate the effectiveness of our proposed processing method in three different channel models: (*i*) AWGN model, where SNR is equal to 10 dB; (*ii*)

Rayleigh fading channel, which is run in the MATLAB software (R2018a) by the *comm.RayleighChannel* function. In this function, it is assumed that our channel model is flat fading with $[0, 1e-4, 2.1e-4]$ path delay, and filtered Gaussian noise is utilized as the fading technique. Moreover, to consider the effect of the mobility of the MSA on the channel model, we assume a flat Doppler spectrum, and; (*iii*) A combination of AWGN and Rayleigh fading channel. To illustrate the superiority of our fusion processing method, we first transmit a signal by the MSA from five different angles, where $\theta_{actual} \in \{10^\circ, 30^\circ, 50^\circ, 70^\circ, 90^\circ\}$. Then, we evaluate the effect of channel model on the angle of the incident signal, obtained based on Eq. (151), and compare the performance of our proposed processing method with MUSIC [185] and Capon [108] algorithms.

Table 5.6 illustrates the impact of AWGN channel model on the AoA estimation framework and compares the efficiency of our proposed processing method with MUSIC and Capon algorithms. The results in Table 5.6 demonstrate the need for processing techniques in order to obtain the accurate angle of arrival. The gap between our processing method, MUSIC, and Capon is, however, negligible as noise is only considered in this scenario. It is worth mentioning that the proposed processing scheme performs significantly better when multipath fading channel is considered in addition to noise, which is an unavoidable consideration in indoor environments.

We investigate the effect of Rayleigh fading channel on the AoA estimation approach in Table 5.7. As it is seen, the proposed processing framework exhibits a higher accuracy when compared with MUSIC and Capon algorithms. This can be explained by noting that the destructive impact of fading channel on the received signal is eliminated by employing ZF Equalizer in our processing method. Moreover, we compare the performance of our proposed processing method with MUSIC and Capon algorithms, where the wireless channel is modelled by the combination of AWGN and Rayleigh fading. With the same arguments as earlier, Table 5.8 illustrates the unreliability of MUSIC and Capon methods against multipath fading while the proposed fusion approach performs reliably.

Figs. 5.19 and 5.20 illustrate rectangular and random trajectories, which are generated by the movement of an MSA in our indoor environment. In this scenario, the wireless channel is modelled by a combination of AWGN and

Table 5.6: Estimated angle of incident signal in AWGN channel model, obtained without processing, by employing the proposed fusion framework, MUSIC and Capon algorithms.

Actual Angle	10°	30°	50°	70°	90°
Without Processing	21.3	59.2	52.2	58.4	82.7
With Processing	11.8	31.0	50.2	71.9	89.3
MUSIC	8.9	30.4	50.1	70.9	89.1
Capon	10.4	30.9	49.1	70.4	90.6

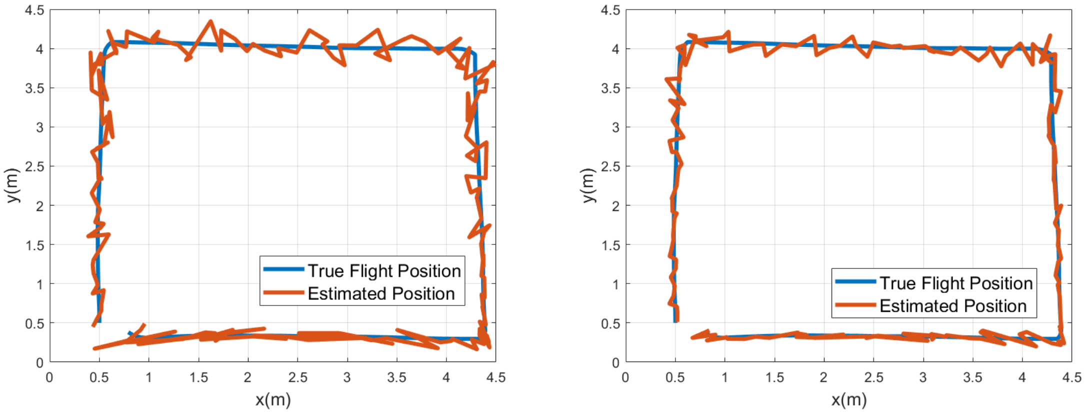
Table 5.7: Similar to Table 5.6 but with Rayleigh Fading channel model.

Actual Angle	10°	30°	50°	70°	90°
Without Processing	0.6	22.1	37.6	57.4	99.5
With Processing	9.7	29.7	49.7	69.7	89.7
MUSIC	20.2	37.7	45.4	63.9	94.3
Capon	42.6	66.3	29.2	22.8	42.6

Table 5.8: Similar to Table 5.6 but with AWGN and Rayleigh Fading channel model.

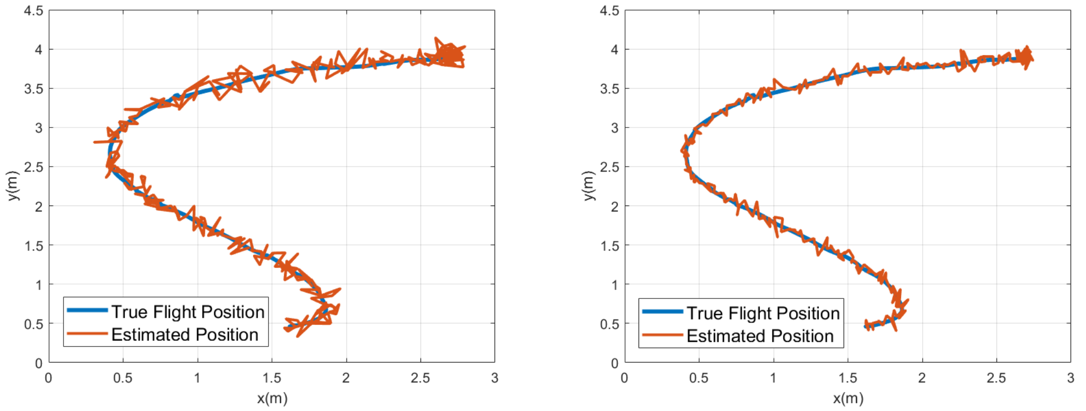
Actual Angle	10°	30°	50°	70°	90°
Without Processing	22.2	55.1	91.8	45.2	118.9
With Processing	8.6	32.1	48.4	83.1	91.3
MUSIC	14.3	19.8	49.3	76.1	106.8
Capon	41.2	45.6	32.4	29.9	32.5

Rayleigh fading channel. As it can be seen from Figs. 5.19 and 5.20, true flight position, which is shown with “blue color”, is obtained by Vicon cameras, and “orange color” illustrates the estimated path according to our proposed fusion framework. To illustrate the impact of noise, we consider two scenarios with the same condition, while in the fist one, depicted in Figs. 5.19(a) and 5.20(a), SNR= 15 dB, and in the second one (Figs. 5.19(b) and 5.20(b)) SNR is set equal to 20 dB. The fluctuation of the estimated paths in Figs. 5.19(a) and 5.20(a) are higher than those obtained in Figs. 5.19(b) and 5.20(b) due to the high level of noise. In both scenarios, however, the estimated paths in the most points are approximately the same with the true flight positions.



(a) (b)

Figure 5.19: Estimated rectangular trajectory based on the true flight position and our proposed processing method for the combination of Rayleigh fading and noise with (a) SNR=15 dB, (b) SNR=20 dB.



(a) (b)

Figure 5.20: Similar to Fig. 5.19 but with random trajectory.

5.4 CNN-based AoA localization

In this Section, we propose an efficient Convolutional Neural Network (CNN)-based indoor localization framework to tackle the aforementioned issues specific to BLE-based settings. We consider indoor environments without presence of Line of Sight (LoS) links affected by Additive White Gaussian Noise (AWGN) with different Signal to Noise Ratios (SNRs) and Rayleigh fading channel. In such real indoor environments, mobile agents and BLE beacons are not always located along the same line, which in turn leads to elevation angle. Although

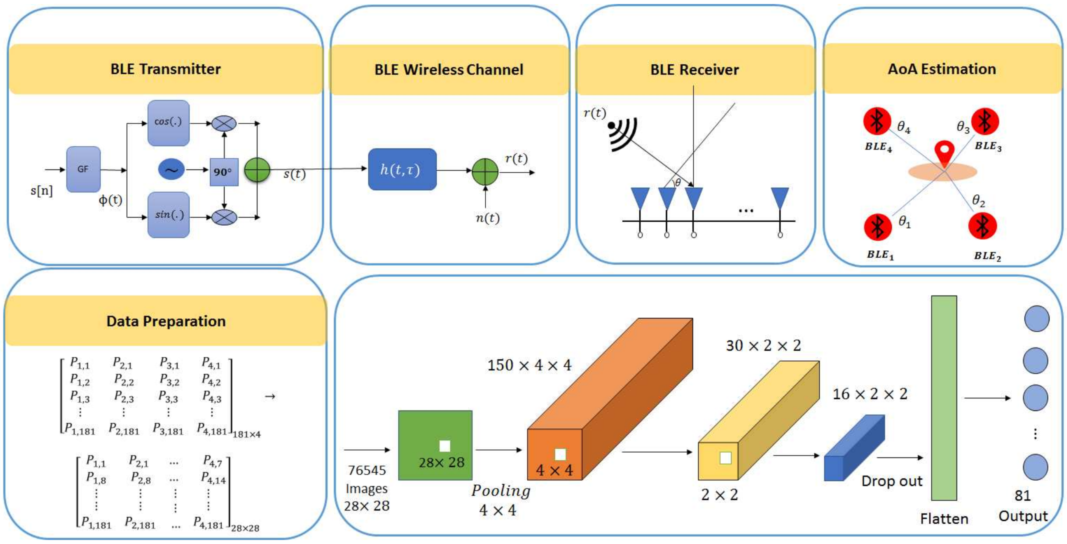


Figure 5.21: Block diagram of the BLE transceiver, wireless channel model, and the proposed CNN-based AoA localization framework.

the azimuth angle of the incident signal is utilized for location estimation, the destructive effect of elevation angle should be considered. Therefore, we generate AoA measurements in a 3-D indoor environment based on the subspace-based angle estimation framework. The raw AoA measurement data, which is obtained in the previous phase, is directly used by the CNN. The input of the CNN architecture is an angle image, where each pixel indicates the spatial spectrum of the AoA measurement.

The proposed CNN-based AoA framework is performed in two phases, i.e., AoA measurement in a 3-D indoor environment, and location estimation based on the CNN algorithm, described below.

5.4.1 AoA measurement in a 3-D Indoor Environment

We consider a subspace-based angle estimation in a 3-D environment, where the incident signal has both azimuth θ and elevation ϕ angles. There are N_e elements in the LAA, receiving the same signal with different phases. By assuming $\lambda = \frac{c}{f_c}$, where $c = 3 \times 10^8 \ m/s$ is the speed of light, the discrete received signal by element e, which is sampled at the discrete time slot m, denoted by $r_e[m]$, is obtained as follows

$$r_e[m] = s^{'}[m]\Theta(\theta, \phi)[m] + n[m], \tag{164}$$

where $\mathbf{\Theta}(\theta,\phi)$ denotes the array vector, defined as follows

$$\begin{aligned}\mathbf{\Theta}(\theta,\phi) = \Big[&\exp(-j\frac{2\pi d}{\lambda}\cos\theta\cos\phi), \\ &\exp(-j\frac{2\pi d}{\lambda}\sin\theta\cos\phi), \exp(-j\frac{2\pi d}{\lambda}\sin\phi)\Big]^T,\end{aligned} \tag{165}$$

where d indicates the space between two consecutive elements of the LAA, which is equal to $\frac{\lambda}{2}$. By assuming M samples in each received signal, we have

$$\mathbf{r} = [r_1[m]\dots r_{N_e}[m]]^T, \tag{166}$$

$$\text{and} \quad \mathbf{s}^{'} = [s_1^{'}[m]\dots s_{N_e}^{'}[m]]^T. \tag{167}$$

Therefore, we can express the received signal in a compact form as

$$\mathbf{r} = \mathbf{\Theta}\mathbf{s}^{'} + \mathbf{n}. \tag{168}$$

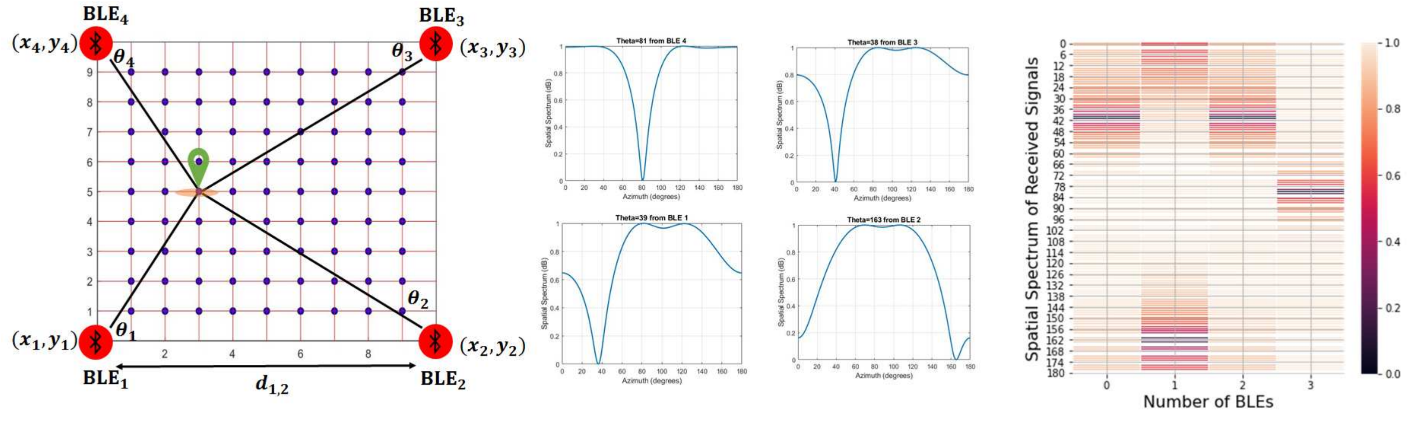
The spatial spectrum function, denoted by $P(\theta,\phi)$, is defined as

$$\mathbf{P}(\theta,\phi) = \frac{1}{\mathbf{\Theta}^H(\theta,\phi)\mathbf{E}_N\mathbf{E}_N^H\mathbf{\Theta}(\theta,\phi)}, \tag{169}$$

where $\mathbf{E}_N$ indicates the noise eigenvectors of the covariance matrix $\mathbf{R} = E[\mathbf{r},\mathbf{r}^H]$. Consequently, the minimum peak of $\mathbf{P}(\theta,\phi)$ illustrates the direction of the incident signal. Given the angle of the signal from at least two BLE beacons with known positions, the location of the mobile agent can be calculated. The coordinate of the BLE beacon b is denoted by (x_b, y_b), and $\theta_{b,n}$ indicates the angle between x-axis and the line between the BLE beacon and the mobile agent. Then, the estimated location of the mobile agent, denoted by $\hat{\mathbf{L}}_n(t) = (x_n, y_n)$, can be estimated using Eq. (152) and (153).

5.4.2 CNN-based Location Estimation

Given the angle of the incident signal, a data-driven approach, which is a combination of a CNN architecture and a subspace-based angle estimation algorithm,



(a)

(b)

Figure 5.22: (a) Experimental data collection of the CNN-based AoA localization framework. (b) An angle image, used as the input of the CNN-based framework.

is designed to track mobile agents during their movements. The proposed architecture, as shown in Fig. 5.21, consists of a series of convolutional layers, pooling layers, fully connected layers, and normalization layers. Each convolutional layer, which applies a convolution operation to the input to extract spatial features, is followed by a pooling layer to down-sample the data to reduce the spatial dimension and the computation time. In each time slot/location, an angle image is provided as the input to the CNN, which is generated by feature matrices constructed based on Eq. (169), i.e., $\mathbf{P}(\theta, \phi, t) = [\mathbf{P}_1(\theta, \phi, t), \ldots, \mathbf{P}_4(\theta, \phi, t)]$. Term $\mathbf{P}_i(\theta, \phi, t)$, for $(i \in \{1, \ldots, 4\})$ indicates the spatial spectrum of the received signal by the i^{th} BLE beacon. This spatial spectrum is reshaped to be an square angle image (see Fig. 5.21). Each angle image is then labeled by the respective ground truth position $\mathbf{L}_n(t) = (x_n, y_n)$. By considering the fact that the angle of the incident signal θ could be valued between $0°$ and $180°$, there are 181 samples in $\mathbf{P}_i(\theta, \phi, t)$, where $\mathbf{P}_i(\theta, \phi, t)$ peak in the corresponding θ. Due to the effects of noise, multi-path, and elevation angle, however, the peak of the spatial spectrum is likely to mismatch the real value θ. Therefore, the goal is to use CNN as a function approximation to estimate the location of the mobile agent from the angle image captured in each time slot. The overall structure of the proposed CNN-based AoA framework is shown in Fig. 5.21. To train the CNN-based AoA framework, the Mean-Squared Error (MSE) is used as the loss

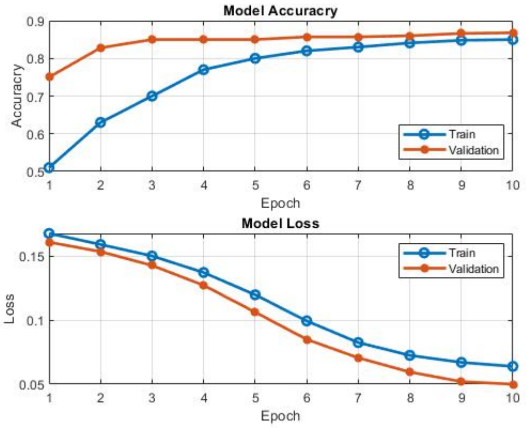


Figure 5.23: Accuracy and loss of the proposed CNN-based AoA scheme.

function $\mathcal{L}(t)$ calculated as

$$\mathcal{L}(t) = \frac{1}{2}\Big(\mathbf{L}_n(t) - \hat{\mathbf{L}}_n(t)\Big)^2, \tag{170}$$

where the estimated location of the mobile agent $\hat{\mathbf{L}}_n(t)$ is defined in Eqs. (152) and (153).

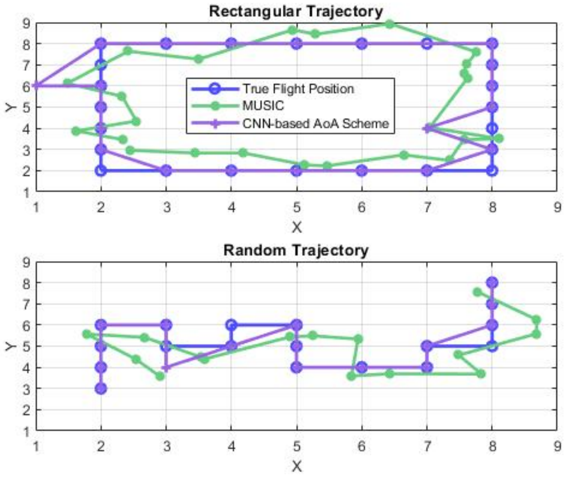


Figure 5.24: Estimated rectangular and random trajectories in a 3-D indoor environment with the presence of Rayleigh fading and noise.

5.4.3 Simulation Results

To evaluate the proposed data driven and BLE-based AoA localization, we used a real experimental testbed consisting of four BLE beacons and Vicon cameras, positioned at the corner of a rectangular indoor area (5×5) m^2 to track a mobile agent and provide ground truth, respectively (see Fig. 5.22(a)). We generated a dataset in three different channel models: (*i*) AWGN model, where SNR is altered between 10 dB and 20 dB; (*ii*) Rayleigh fading channel, which is implemented in MATLAB (R2020a) by the *comm.RayleighChannel* function, and; (*iii*) A combination of AWGN and Rayleigh fading channel in a 3-D indoor environment, divided into 81 square zones with dimension of (0.5×0.5) m^2. Although elevation angle is not considered for location estimation, it has a destructive impact on the special spectrum of the AoA measurement, leading to a specific error in the users' tracking. For this reason, it is assumed that the incident signal is received by different elevation angles. In the training phase, $76,545$ (angle image, location) training points are utilized corresponding to 81 random locations within the area. As it can be seen from Fig. 5.22(b), angle images contain 724 sets of AoA measurements generated by the subspace-based algorithm. This original angle image is of size 4×181 and is reshaped to be an square angle image of size 28×28 (zero padding is performed to fit the CNN model preventing data loss). In the next phase, we use $15,309$ images as the validation set and $10,206$ for our test set, which are all previously unseen and randomly chosen. For the CNN model, we feed our data to three 2-D convolution layers, each with 196 filters with 4 kernel size. We added maximum pooling layer and utilized sigmoid as an activation function. Finally, within the dense layers, we used linear function as the activation function.

Fig. 5.23 illustrates the accuracy and the loss of the proposed CNN-based AoA framework, respectively. As it can be seen from Fig. 5.23, increasing the number of epoches increases the model accuracy and decreases the model loss, which shows that the model is well trained. The proposed CNN-based AoA framework estimates/tracks mobile agents with 87% accuracy in the presence of noise, Rayleigh fading, and elevation angle. Fig. 5.24 compares the rectangular and random trajectories, estimated by our proposed CNN-based AoA framework, the 3-D MUSIC and the true flight position, obtained by Vicon cameras. In both scenarios, the estimated paths by the proposed CNN-based AoA framework in the most points closely follows that of the true flight positions.

5.5 Deep Q-Learning-based Energy-optimized LoS/NLoS Connection

In this Section, we aim to address the unbalanced energy consumption issue, the location error caused by the NLoS connections, and the time varying behavior of indoor environments. In this context, we investigate an autonomous and homogeneous indoor localization framework including fixed and known location UWB beacons covering the area of interest. The goal is to monitor/track user's movement with high accuracy in the presence of NLoS condition. Due to the limited battery life of UWB beacons and the computational complexity of the localization phase, it is essential to minimize the number of cooperating nodes for localization. The novel approach we are taking here is to train the mobile user to be localized through the optimal UWB beacons with LoS links, while maintaining a balance between the remaining battery life of all beacons. In summary, this work makes the following key contributions:

- Due to the reflective obstacles such as walls and human body, indoor environments suffer from NLoS connections, which degrade the location accuracy exponentially. Without applying complex NLoS mitigation methods, we introduce an autonomous localization framework, where the mobile user is trained to be localized by UWB beacons with LoS conditions at each time/location.
- In such a scenario that a set of UWB beacons with LoS condition mostly contribute to localization, their batteries would be fully drained in a short period of time. It is, therefore, imperative to consider the remaining battery life of beacons as a selection criteria. Analytical anchor node selection frameworks are unable to cope with the dynamic nature and time-varying behavior of indoor environments. We, therefore, target development of an adaptive anchor selection framework efficiently copes with the dynamic nature of indoor environments. Despite the surging interest in the anchor

node selection frameworks, there is no Reinforcement Learning (RL)-based framework concerning a trade-off between the remaining battery life of UWBs and localization accuracy. The proposed DQLEL framework addresses this gap via a RL-based formulization with the goal of maintaining a balance between the location error by selecting UWB beacons with the LoS condition and the remaining battery life of UWB beacons.

The effectiveness of the proposed DQLEL framework is evaluated through comprehensive simulation studies in terms of the location error, the mean deviation of UWB's remaining battery life, the link condition, and the cumulative rewards. Simulation results illustrate the efficiency of the proposed DQLEL scheme in comparison to its state-of-the-art counterparts over all the aforementioned aspects.

5.5.1 System Model and Problem Description

Our research scenario is a corridor inside a building (e.g., an office or a hotel building), consisting of N rooms where each room is equipped with a synchronized UWB beacon as the receiver node, denoted by UWB_i, $i = 1, \ldots, N$. We also consider a mobile user as the transmitter node, who randomly moves within the corridor. The reception range (assumed to be equal) and the battery life of UWB beacons at time slot t are denoted by R_i and $B_{i,t}$, for $(1 \leq i \leq N)$, respectively. Due to the existence of reflective obstacles such as walls and movement of other users, such indoor environments suffer from multipath, shadowing, and pathloss effects, which are known by NLoS links. Therefore, the transmission link between UWB beacons and the user at each location can be either LoS or NLoS. Despite the ToA approach, where a strict synchronization is required between the transmitter and the receiver, only synchronization between UWB beacons is required in the TDoA. By relaxing the assumption that all UWB beacons are synchronized, our objective is to train the user to be localized through an optimal set of UWB beacons with LoS links, without draining the battery of certain UWB beacons. In this Subsection, we present the wireless signal model of the IR-UWB standard and formulate the transmitted signal, wireless channel, and the received signal to extract the user's location through

Table 5.9: List of Notations.

Notation	Description	Notation	Description
N	Total number of UWB beacons	P_{COR}	Correlator branch power consumption
N_u	Number of available beacons	P_{ADC}	ADC power consumption
R_i	Reception range of UWB$_i$	P_{LNA}	LNA power consumption
$B_{i,t}$	Battery life of UWB$_i$ at time slot t	P_{VGA}	VGA power consumption
τ_i	First peak of the estimated CIR of UWB$_i$	P_{GEN}	Pulse generator power consumption
(x_t, y_t)	User's location at time slot t	P_{SYN}	Synchronization power consumption
(x_i, y_i)	Location of UWB$_i$ at time slot t	P_{EST}	Channel estimator power consumption
P_r	Power consumption of an UWB receiver	E_L	Payload energy consumption
ρ_t	Pulse coefficient	E_O	SP and PHR energy consumption
ρ_c	Symbol repetition parameter	E_{RX}	Energy consumption for receiving one packet
ρ_r	Demodulation parameter	E	Total energy consumption in reception session
L_{SP}	Number of SP symbols	E_{IPS}	Energy consumption during T_{IPS}
L_L	Number of bits in the payload	E_{ACK}	Acknowledgement power consumption
L_{PHR}	Number of bits in the PHR	T_{tr}	Transient session
R_{base}	Fixed base data rate	T_{on}	Time duration for receiving one packet
R_c	Coding rate	T_{onL}	Time duration of the payload
N_p	Coding parameter	T_{PHR}	Time duration of the PHR
T_{ACK}	Acknowledgement time	T_{SP}	Time duration of the SP
T_{IPS}	Inter packet space time duration	M	Number of RAKE fingers

the time information. Afterwards, the energy consumption model of the UWB beacons as the receiver nodes will be introduced. A summary of the notations used hereinafter is provided in Tables 5.9.

IR-UWB Wireless Signal Model

In the IR-UWB technology, sequences of short time-domain impulses transmit over a high-bandwidth radio spectrum resulting in an improvement in data rate and localization accuracy for short-range communication. We utilize the Time Hopping (TH) technique as one of the efficient Code Division Multiple Access (CDMA) schemes that guarantees the optimal pair of codes with high probability by increasing the number of chips per symbol. Considering the Pulse Amplitude Modulation (PAM) method, the transmission signal from mobile user u is given by

$$s_u(t) = \sum_{n=-\infty}^{\infty} p_u(n) \sum_{s=0}^{N_c-1} c_u(s) w(t - nT_s - sT_c - \theta_u) \tag{171}$$

where T_s denotes the symbol time, when there are N_c chips with duration of T_c in each symbol. Terms $w(t)$ and θ_u represent the normalized impulse and the time asynchronism which has a uniform distribution within $[0, T_s]$, respectively. Finally, terms $p_u(n) \in \{-1, 1\}$ and $c_u(s) \in \{0, 1\}$ are the Independent and Identically Distributed (IID) information symbols, and the multiple access code of the mobile user u, respectively. In indoor environments, UWB beacons, typically, receive a number of phase delayed and power attenuated versions of the transmitted UWB signal from different users, affected by Additive White Gaussian Noise (AWGN). The received signal by UWB_i, denoted by $r_i(t)$, therefore, can be expressed as

$$r_i(t) = \sum_{u=1}^{N_u} \sqrt{P_u} \sum_{k=1}^{N(t)} \rho_{u,k}(t, \tau) s_u(t - \tau_{u,k}(t) - \tau_i) + n(t), \tag{172}$$

where $N(t)$ represents the number of detachable paths, and N_u is the number of users in the experimental indoor environment. Terms $\rho_{u,k}(t, \tau) = \beta_{u,k}(t, \tau) \exp(j\Phi_{u,k}(t, \tau))$ and $\tau_{u,k}(t)$ denote the attenuation and the delay of the k^{th} path, where $\beta_{u,k}(t, \tau)$

and $\Phi_{u,k}(t,\tau)$ represent the path amplitude and phase, respectively. Term τ_i represents the ideal time information, which is equal to $\tau_i = d_i/c$, where d_i is the distance between mobile user u and UWB_i, and c is the speed of light, about 3×10^8 m/s. Term $n(t)$ represents the AWGN channel, which is modelled by $n(t) \sim \mathcal{N}(0, \sigma^2)$. Finally, term P_u denotes the information of random captured powers. Note that, $\beta_{u,k}(t,\tau)$ is the Nakagami-m random variable as the small-scale channel coefficient of the link between the user and the UWB beacon, where m indicates the degree of fading severity. By assuming that there is a strong LoS path between the transmitter and the receiver, the radio propagation is modelled by Rician fading channel model, otherwise, it would be Rayleigh fading. More specifically, $m = 1$ represents the Rayleigh fading, while $m > 1$ is for Rician channel model. Under the Nakagami-m fading assumption, the small-scale channel gain denoting by $|\zeta_k|^2 \sim \Gamma(m, 1/m)$ is a normalized independent and identically distributed (i.i.d.) Gamma random variable. The Probability Density Function (PDF) of the power fading is expressed as

$$f_{|\zeta_k|^2}(x) = \frac{m^m}{\Gamma(m)} x^{m-1} e^{-mx}, \quad x > 0 \tag{173}$$

where $\Gamma(\cdot)$ is the Gamma function. In an UWB positioning system, the user's location can be obtained from the estimated CIR, which is expressed as

$$h(t,\tau) = \sum_{u=1}^{N_u} \sum_{k=1}^{N(t)} \rho_{u,k}(t,\tau) \delta(t - \tau_{u,k}(t) - \tau_i), \tag{174}$$

where the first maximum peak of the estimated CIR in the LoS condition is associated with the delay of the first path. TDoA scheme determines the location of users by calculating the time difference between the transmitted signal from the user and the signals received by at least two UWB beacons in a 2-D indoor environment. In such a scenario that the corresponding user is located in the reception range of UWB_i and UWB_j, the TDoA information, denoted by T_{ij}, is given by

$$T_{ij} = \tau_i - \tau_j = \frac{\sqrt{\left(x_t - x_i\right)^2 + \left(y_t - y_i\right)^2} - \sqrt{\left(x_t - x_j\right)^2 + \left(y_t - y_j\right)^2}}{c}, \tag{175}$$

where τ_i and τ_j represent the first peak of the estimated CIR of UWB_i and UWB_j. In addition, (x_t, y_t), (x_i, y_i), and (x_j, y_j) denote the locations of the corresponding user at time slot t, UWB_i, and UWB_j, respectively. This completes the description of the UWB wireless signal model. Next, we introduce the RL background.

IR-UWB Energy Consumption Model

Due to the limited battery life of UWB beacons as the receiver nodes, it is crucial to extend the lifetime of beacons. Toward this goal, first we present the energy consumption of UWB beacons [187]. The power consumption of an IR-UWB receiver P_r is expressed as

$$P_r = P_d + P_n, \tag{176}$$

where P_d and P_n represent the circuit components' power consumption associated with the detection scheme, and the rest of the components, respectively. In such a case, P_d and P_n are calculated as follows

$$\begin{aligned} P_d &= MP_{COR} + \rho_c P_{ADC} + P_{LNA} + P_{VGA}, & (177)\\ \text{and} \quad P_n &= \rho_r (P_{GEN} + P_{SYN} + P_{EST}). & (178)\end{aligned}$$

where P_{COR}, P_{ADC}, P_{LNA}, and P_{VGA} denote the power consumption of the correlator branch including a mixer and an integrator, the Analog-to-Digital Converter (ADC), the Low Noise Amplifier (LNA), and the Variable Gain Amplifier (VGA), respectively. Term M denotes the number of RAKE fingers at the receiver side, which is assumed to be 1. Symbol repetition scheme, including Hard Decision (HD) combining and Soft Decision (SD) combining is denoted

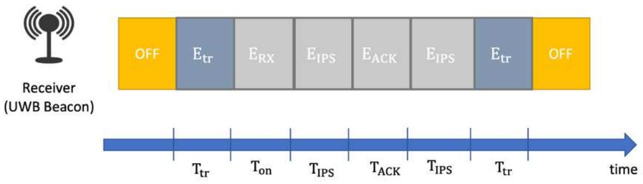


Figure 5.25: Energy consumption of UWB beacons during one packet reception.

by ρ_c, where $\rho_c = 1$ is used for SD and $\rho_c = 0$ is for HD combining. In addition, P_{GEN}, P_{SYN}, and P_{EST} represent the power consumption associated with the pulse generator, the synchronizer, and the channel estimator, respectively. Finally, ρ_r is an structural parameter, where $\rho_r = 1$ is related to the coherent demodulation while $\rho_r = 0$ is for noncoherent demodulation.

The IR-UWB data packet is constructed by: (i) Synchronization Preamble (SP); (ii) PHY-Header (PHR), and; (iii) Payload components. Therefore, the energy consumption of receiving one data packet consists of the energy consumed on the payload, denoted by E_L, and the energy consumption associated with delivering the SP and PHR, denoted by E_O, where E_O is expressed as

$$E_O = P_r\big(T_{SP} + T_{PHR}\big) = P_r \underbrace{\frac{(L_{SP} + \dfrac{L_{PHR}}{R_c})}{R_{base}}}_{T_O}, \tag{179}$$

with the assumption that there are L_{SP} symbols in the SP, and L_{PHR} bits in the PHR, respectively. The time duration of the PHR, and the SP are represented by T_{PHR}, and T_{SP}, respectively. Term $R_c = 1/N_p$ is the coding rate, where N_p is the coding parameter, which must be an odd number. Finally, R_{base} is the fixed base data rate. The energy consumption for receiving the payload, containing L_L information bits, denoted by E_L, is expressed as

$$\begin{aligned} E_L &= \rho_t(MP_{COR} + \rho_c P_{ADC} + P_{LNA} + P_{VGA})T_{onL} \\ &+ \rho_r(P_{GEN} + P_{SYN} + P_{EST})T_{onL}, \end{aligned} \tag{180}$$

where ρ_t represents the pulse coefficient, and T_{onL} denotes the time duration of the payload. The energy consumption to receive one packet within time T_{on} is

$E_{RX} = E_O + E_L$, which can be expressed as

$$T_{on} = T_{SP} + T_{PHR} + T_{onL} = \frac{(L_{SP} + \frac{L_{PHR}}{R_c})}{R_{base}} + \frac{L_L}{R_b R_c}, \tag{181}$$

where R_b denotes the bit rate. Since our focus is to manage the energy consumption of UWB beacons, the amount of energy consumed by users as the transmitter is not modeled. As it can be seen from Fig. 5.25, the total energy consumed by UWB beacons during each packet reception session can be expressed as

$$E = 2E_{tr} + E_{RX} + 2E_{IPS} + E_{ACK}, \tag{182}$$

where $E_{IPS} = \rho_r P_{SYN} T_{IPS}$ and $E_{tr} = \rho_r P_{SYN} T_{tr}$. In this case, T_{tr} represents the time that the UWB beacon switches from the sleep state to an active state for receiving a data packet, and T_{IPS} denotes the inter packet space time duration. Note that the energy consumption E only represents the UWB energy consumption for a single packet reception session [187], which does not include the energy consumed by the UWB beacon for processing the received signal to extract the desired information, i.e., LoS/NLoS connection. Furthermore, we would like to note that there is no need for checking channel condition of UWB beacons in the vicinity of the mobile device. Moreover, E_{ACK} is calculated as

$$E_{ACK} = (L_{SP} + \frac{L_{PHR}}{R_c})E_p + P_{SYN} T_{ACK}, \tag{183}$$

where $T_{ACK} = T_O$ is the time duration when the UWB beacon listens for an ACK acknowledgement from the corresponding user. This completes the description of the UWB power consumption model. Next, we introduce our proposed DQLEL UWB node selection framework.

5.5.2 DQLEL UWB Node Selection

DQLEL Framework

As stated previously, extracting time information from NLoS links leads to an increase in the location error. At the same time, extracting time information

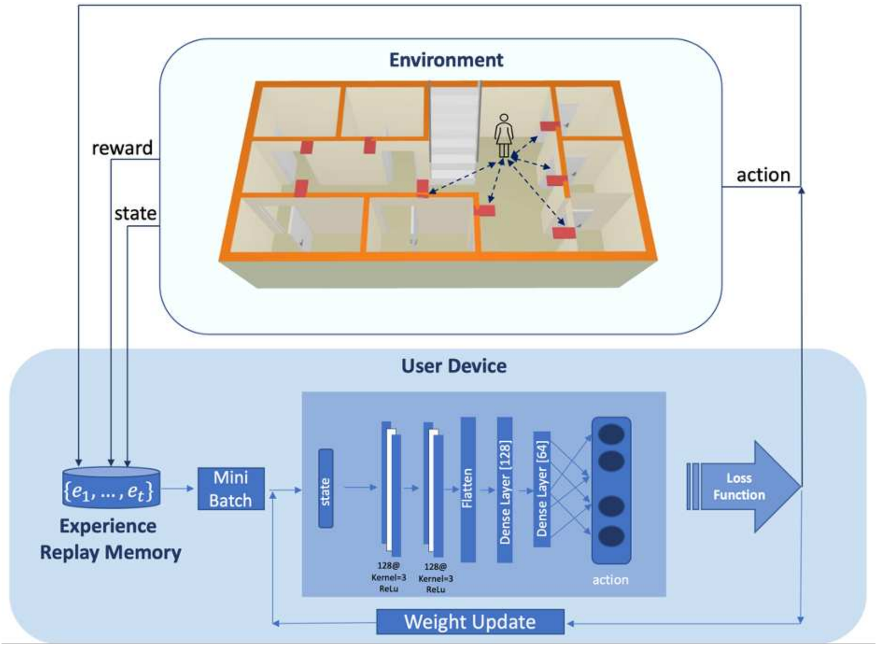


Figure 5.26: The block diagram of the proposed DQLEL framework.

from all beacons is inefficient from energy consumption perspective. The proposed DQLEL framework capitalizes on these facts aiming to autonomously train the mobile user to find energy-optimized UWB beacons with LoS links at each location. Considering the fact that applying a complex RL model is energy consuming, it is commonly assumed that UWB beacons transmit their sensory data to the central server [186] to perform the DQLEL framework. As it can be seen from Fig. 5.26, the DQLEL framework consists of the following main components:

(i) Agent: The mobile user acts as the agent within the DQLEL framework and interacts with the environment based on a set of given actions defined below.

(ii) Action-Space: The action space in the DQLEL framework refers to the selection of a set of UWB beacons to determine the location of the mobile user. Therefore, the cardinality of the action space N_a is defined as follows

$$N_a = \frac{N_u!}{(N_u - N_b)!\ N_b!}, \tag{184}$$

where N_u and N_b denote the total number of UWB beacons in the vicinity of the user, and the number of active beacons required for the localization purposes, respectively. Note that the main goal of our work is to use the minimum number of active UWB beacons among all available ones to not only improve the localization accuracy, but also mitigate excessive energy consumption of transmitting/receiving beacons. The minimum number of required active UWB beacons varies between 2 and 4, depending on the type of the time estimation algorithm such as ToA or TDoA, and the dimension of the proposed environment, i.e., 2D and 3D. For instance, in 2D and 3D positioning, ToA uses at least three and four UWB beacons and requires three and four measurements of distances from one node to another to localize a mobile device, respectively. On the other hand, the number of UWB beacons required for TDoA is one less than that of the ToA scheme, therefore, the location of a mobile device can be determined by the intersection of two and three hyperbolas in 2D and 3D positioning, respectively. Utilization of three UWB beacons provides higher location accuracy for the 2-D TDoA approach, especially in some specific configurations where the localization accuracy of two UWB nodes may not meet the desired accuracy. However, in such configurations where the DRL model should be able to localize the mobile device by 2 or 3 beacons, an adaptive DRL model with two different modes is required, i.e., one action-space associate with a set of 2 UWB beacons, and another one related to the set of 3 UWB beacons, which in turn significantly increases the cardinality of the action-space. By increasing the action-space, the learning time of the DQLEL framework, which is required for converging to the optimal policy will be extended. As the first step towards development of a fully autonomous agent for UWB-based indoor localization, therefore, we considered one action-space mode, where the number of UWB beacons in each action is $N_b = 2$. The selected action is a vector, denoted by $\mathbf{a} = [a_i, a_j]$, where a_i and a_j represent UWB_i and UWB_j, respectively. One fruitful direction for future research is development of an adaptive DRL localization framework capable of increasing the number of active UWB beacons in some specific configurations, e.g., when the localization accuracy of two UWB nodes fails to meet the desired accuracy.

(iii) State-Space: The action is selected based on the current state of the system $\mathbf{s}_t$ at time slot t. Each state consists of the user's location (x_t, y_t), and the UWB beacons' battery life $\mathbf{B}_t$. More specifically, $\mathbf{s}_t$ consists of the following components:

- (x_t, y_t): Location of the user. To evaluate performance of the DQLEL framework in terms of the location error, it is assumed that a priori knowledge of the exact user's location is available, which is a common assumption in Machine Learning (ML)-based indoor localization frameworks [188, 189]. Following Reference [216], the indoor environment is discretized into $N_l = N_x \times N_y$ points, where x_t and y_t at time slot t are obtained as

  $$\begin{aligned} x_t &= x_{t-1} + x_m, \qquad 0 \le x_t \le N_x && (185) \\ \text{and } y_t &= y_{t-1} + y_m, \qquad 0 \le y_t \le N_y. && (186) \end{aligned}$$

  where (x_{t-1}, y_{t-1}) denotes the user's location at time slot $t-1$, and $x_m, y_m \in \{-1, 0, 1\}$ indicate the user's movement [216] and are selected randomly.
- $\mathbf{B}_t$: Vector $\mathbf{B}_t = [B_{1,t}, \ldots, B_{N_u,t}]$ illustrates the battery life of UWB_i, for $(1 \le i \le N_u)$. By taking the action $\boldsymbol{a}$, the battery life of two involving beacons are updated as

  $$B_{i,t} = B_{i,t-1} - E, \tag{187}$$

  where E denotes the energy consumption of UWB beacon in the reception session, obtained by Eq. (182), will be described shortly in Appendix-A.

(iv) Reward: The reward function in the proposed DQLEL framework is defined in such a way that the user selects UWB beacons results in higher location accuracy, while maintaining the load balance between UWB beacons. There are several key parameters affecting the location accuracy, which can generally be categorized into two groups, i.e., (i) Infrastructure-based factors such as UWB beacons installation, and the number of active UWB beacons in the vicinity of a mobile user, and; (ii) Beacon-based factors [190, 191]. Despite the fact that infrastructure-based factors have a substantial impact on the location accuracy, most existing research studies [192, 216] assumed a pre-installed indoor setting with a pre-determined number of beacons. The second group, the beacon-based factors, include channel condition (the LoS/NLoS connection), data rate, and signal modulation to name a few. Data rate is the number of transmission data from a mobile user to the UWB beacon to estimate the current location. Although a higher data rate improves the location accuracy, it consumes more energy, therefore, a trade-off between these two metrics should be considered to determine the optimal data rate. Moreover, signal modulation has a significant impact on the quality of the UWB transmitting signal, which can be resulted in higher location accuracy in the presence of noise and interference. To better understand the influence of the channel condition on the location accuracy, we consider a pre-defined data rate and signal modulation, which is a common assumption in indoor localization studies [101,103,192,216]. By taking an action, the estimated location of the user is calculated. Then, the location error at time slot t, denoted by $\mathcal{ER}_t$, is calculated as

$$\mathcal{ER}_t = \sqrt{(x_t - x_{es,t}^{(i,j)})^2 + (y_t - y_{es,t}^{(i,j)})^2}, \tag{188}$$

where (x_t, y_t) is the exact user's location at time slot t and $(x_{es,t}^{(i,j)}, y_{es,t}^{(i,j)})$ denotes the estimated user's location, which is obtained by UWB_i and UWB_j. Note that selecting UWB beacons with LoS links leads to a remarkable reduction in the value of $\mathcal{ER}_t$.

To provide a fairness UWB connection scheduling, the mean deviation of UWB energy consumption is introduced as the load balance metric. In this regard, the deviation of remaining battery of UWB beacons at time slot t, denoted by $\mathcal{MD}_t$, is the distance between the battery of all beacons at time slot t and the average one, given by

$$\mathcal{MD}_t = \sqrt{\frac{1}{N_u - 1} \sum_{i=1}^{N_u} \frac{(B_{i,t} - \overline{B}_t)^2}{\overline{B}_t^2}}, \tag{189}$$

where $\overline{B}_t$ represents the average UWB battery life at time slot t. The reward function $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is defined as

$$\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} \dfrac{1}{\mathcal{MD}_t \mathcal{ER}_t}, & \mathcal{MD}_t \leq \mathcal{MD}_{th}, \mathcal{ER}_t \leq \mathcal{ER}_{th}, \\ -\mathcal{MD}_t \mathcal{ER}_t, & \text{o.w.} \end{cases}, \tag{190}$$

where $\mathcal{MD}_t$ denotes the deviation of remaining battery of UWB beacons at time slot t, $\mathcal{ER}_t$ denotes the location error at time slot t, and term $\mathcal{E}_{th}$ is a pre-defined threshold value for the maximum acceptable location error [192]. After selecting two UWB beacons with the largest reward function $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, the connection information associated with the corresponding action and state are stored in the memory replay of the proposed DQLEL model. Due to the infinite state-action space, we use the CNN architecture as a non-linear approximator in the Q-learning model to approximate the Q-value of each state-action pair.

In such scenarios with a high number of state-action pairs, the computational cost of conventional Q-learning algorithms is significantly high. Therefore, Deep Q Learning (DQN) models are, typically, used where instead of storing expected rewards associated with each state-action pair in a Q-table, a Deep Neural Network (DNN) model is used to select the actions according to the agent's current state [193]. In complex problems such as the one at hand, several information sources (such as position of the mobile user, the set of available UWB beacons, and their battery life at each time slot/location) are simultaneously required to perform the action selection task. CNN architecture [194] is an attractive solution to extract the relevant features from this pool of information. CNN-based architecture uses convolutional kernels to compress the state-space and extract temporal correlations between the current state of a mobile user and previous state-action pairs. Within the incorporated CNN architecture, weights are shared between the episodes, which leads to a considerable reduction in the computational complexity. Since the state space is a vector, the CNN module used in the DQLEL framework consists of 1-dimensional convolutional layers. More precisely, there are two 1-dimensional convolutional layers with 128 filters, each with the size of 3 and with the ReLU activation function. There are also two Fully Connected (FC) layers, where the first one consists of 128 ReLU units and the latter has 64 ReLU units. The state of the DRL model is used as the input of the CNN, consisting of the user's location (x_t, y_t), and the UWB beacons' battery life $\mathbf{B}_t$, where the number of UWB beacons are N_u. The input size, therefore, is equal to the size of the state-space, which is equal to $(2 + N_u)$, where 2 is associated with the user's location (x_t, y_t). Considering the fact that the main goal of the DQLEL framework is to determine the optimal set of UWB beacons for localization in each time stamp/location, the output size of the CNN is equal to the size of the action space N_a. Moreover, the activation function of the output layer is softmax. To maintain a trade-off between the exploration and the exploitation of the DQLEL framework, a variable ϵ is assumed for the ϵ-greedy action selection policy. The maximum value of ϵ, denoted by ϵ_{max}, is equal to 1, gradually decreasing with time by $\Delta\epsilon = \dfrac{\epsilon_{max} - \epsilon_{min}}{N_{epoch}}$ until a steady state is reached, where $\epsilon_{min} = 0.01$, and N_{epoch} is the total number of epochs, equal to 500 in this work. In such a scenario, the random action $\boldsymbol{a}_t$ is selected at time slot t with the probability of ϵ.

A replay memory is used to retrain the CNN model for previously observed state-action pairs and their corresponding rewards. Therefore, β number of state-action pairs at time slot t, denoted by $\boldsymbol{\phi}_t = [\boldsymbol{s}_{t-\beta}, \boldsymbol{a}_{t-\beta}, \ldots, \boldsymbol{a}_{t-1}, \boldsymbol{s}_t]$, are used as the input of the CNN to estimate $Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_t)$, where ξ_t denotes the filter weight at time slot t. The experience memory pool is denoted by $D = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t\}$, where $\boldsymbol{e}_t = (\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1})$. To update the weight parameter ξ_t using the Stochastic Gradient Descent (SGD) method, the state sequence in replay buffer $\mathbf{e}_m$ is selected at random. Given the value of ξ_t, the goal is to obtain the optimal action in each time slot, which is obtained by minimizing the following loss function

$$\mathbb{L}(\xi_t) = \mathbb{E}_{\boldsymbol{\phi}_t}, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t+1} \left[\Big(Q_T - Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t|\xi_{t+1}) \Big)^2 \right], \tag{191}$$

where Q_T is the target optimal Q-function, expressed as

$$Q_T = r_t + \gamma \max_{\boldsymbol{a}_t^{'}} Q(\boldsymbol{\phi}_{t+1}, \boldsymbol{a}_t^{'}|\xi_{t-1}). \tag{192}$$

According to the ϵ-greedy algorithm, the best action $\boldsymbol{a}_t^*$ for the state $\boldsymbol{s}_t$ is chosen from the set of Q-functions with the probability of $(1 - \epsilon)$ as follows

$$\boldsymbol{a}_t^* = \arg\max_{\boldsymbol{a}_t^{'}} Q(\boldsymbol{\phi}_t, \boldsymbol{a}_t^{'}). \tag{193}$$

Given the action $\boldsymbol{a}_t^*$, two UWB beacons associated with action $\boldsymbol{a}_t^*$ are involved to track the user's location at time slot t. Eventually, the new experience $\{\boldsymbol{\phi}_t, \boldsymbol{a}_t, r_t, \boldsymbol{\phi}_{t-1}\}$ is stored in the replay memory by the agent.

Table 5.10: List of Parameters.

Notation	**Unit**	**Value**	**Notation**	**Unit**	**Value**
L_{SP}	symbols	1024	P_{SYN}	mW	30.6
L_{PHR}	symbols	16	P_{ADC}	mW	2.2
N_p	-	$\{1, 3, 5, \ldots, 15\}$	P_{GEN}	mW	2.8
E_p	pJ/pulse	4.5	P_{LNA}	mW	9.4
R_{base}	Mbps	1	P_{EST}	mW	10.08
P_{COR}	mW	10.08	P_{VGA}	mW	22

Computational Complexity

We compute the computational complexity of the CNN as the learning method of the proposed DQLEL framework. Generally speaking, the computational complexity of a CNN model with $\mathcal{N}_l$ number of convolutional layers, where each layer includes F_l filters with size $W_l^f \times L_l^f$, is

$$\mathcal{N} = \sum_{l=1}^{\mathcal{N}_l} F_{l-1} W_l^f L_l^f F_l W_l^o L_l^o, \tag{194}$$

where F_{l-1} and F_l represent the number of input channels and filters corresponding to the l^{th} layer, respectively. In addition, W_l^o and L_l^o denote the width and the length of the output, calculated as follows

$$W_l^o = \frac{W_{l-1}^o - W_l^f + 2P_l}{S_l} + 1, \tag{195}$$

$$\text{and } L_l^o = \frac{L_{l-1}^o - L_l^f + 2P_l}{S_l} + 1, \tag{196}$$

where S_l and P_l represent the size of stride and padding layers of the l^{th} layer, respectively. Moreover, there are $\mathcal{N}_{fc}$ number of fully connected layers for estimating the Q-value associated with each action. Considering that the pooling and fully connected layers only take up $5-10\%$ of the computational time [194], their impact on the computational complexity of the CNN can be negligible. Accordingly, the value of $F_0 \times W_0^o \times L_0^o$ is equal to βl_s in the DQLEL framework, where β is the temporal memory depth, and l_s represents the length of the state that is equal to $(2 + N_u)$.

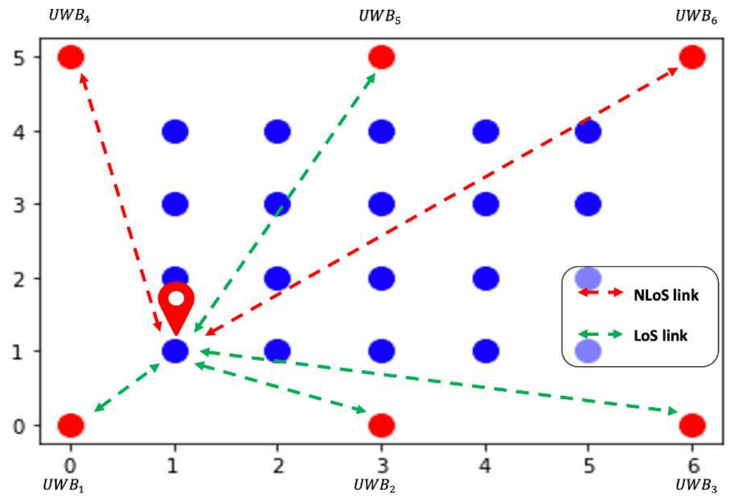


Figure 5.27: A typical sub-area in an experimental indoor environment consisting of 6 UWB beacons for UWB node selection.

5.5.3 Evaluations and Simulation Results

To evaluate the effectiveness of the proposed DQLEL scheme, we consider an experimental indoor testbed with the size of (60×50) m^2. Indoor environments, such as hotels and office buildings, can be divided into several non-overlapping sub-areas [106, 216], where the mobile user is localized in each sub-area through all that sub-area's UWB beacons. Due to the limited transmission area of UWB beacons, after transmitting an UWB signal by the user at each location, the transmitted signal can be received by N_u number of UWB beacons. Without loss of generality, we assume a rectangular configuration for each sub-area in the indoor testbed, which is a common experimental set-up in pre-installed indoor localization applications [195, 196]. Fig. 5.27 illustrates a typical sub-area consisting of six UWB beacons, located at a (6×5) m^2 rectangular indoor environment. Following Reference [216], the sub-area is divided into (5×4) square zones with dimension of (1×1) m^2. Although the location resolution, i.e., the number of discretized points in the indoor environment N_l, is proportional to the location accuracy, it also results in higher state-space, complexity, and the respond-time. Therefore, there should be a trade-off between the location resolution and the respond-time of the learning model. Despite the recent DRL-based localization works [192, 216], where the environment is divided into a grid of (5×5) m^2 and (3×3) m^2 cells, respectively, we assume higher resolution of (1×1) m^2 to improve the location accuracy. Furthermore, we use CNN as the learning model in the DQLEL framework instead of Multi Layer Perceptron (MLP) [192, 216] to avoid sacrificing the respond-time of the learning model, resulting in the faster convergence. For convergence of the DQLEL model, i.e., minimizing the loss function, we assumed that the DRL model is performed over 500 epochs (each epoch consists of several measurements), in which each epoch ends when the battery life of at least one UWB beacon is completely drained, known as the *game over* in our DRL network.

The mobile user moves across the environment in 8 directions according to Eqs. (185) and (186) based on the random walk model [216]. At each time slot, the mobile user is located at the center of each zone [216], shown by a blue point in Fig. 5.27. The channel condition of the received signal depends on the LoS/NLoS of the signal. To consider effects of obstacles on the received signal, the channel condition of UWB_i, for $(1 \leq i \leq N_u)$, at each location is determined randomly. Two out of N_u beacons will be selected for localization. Table 5.10 illustrates the list of other parameters [187] used for running the experiments. Following Reference [187], the optimum packet length and the minimum UWB energy consumption depend on the distance between the mobile user and UWB beacons. Considering the fact that the optimal packet length is approximately $1,500$ Kbit for the transmission range of upto 10 m [187], the energy consumption E in the DQLEL framework is constant. Please refer to Appendix A for a complete discussion on the IR-UWB energy consumption model.

Effectiveness of the DQLEL Framework

To model the LoS/NLoS communications in this work, it is assumed that there are several obstacles in the experimental indoor environment. Initially, the experimental indoor testbed is set up by considering a variety of random obstacles across the environment, resulting in at least two NLoS connections being established at each point in each sub-area. Because this study is conducted using simulations, obstacles are considered to be distributed randomly throughout the environment, resulting in the designation of some communications as LoS

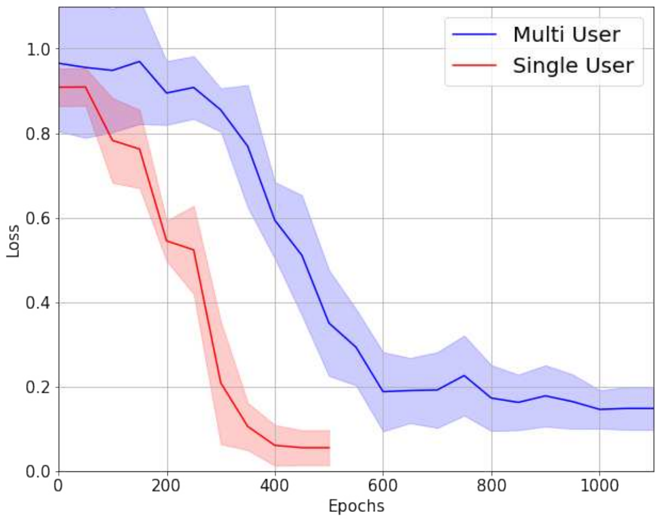


Figure 5.28: The convergence of the DQLEL framework in single and multi user environments.

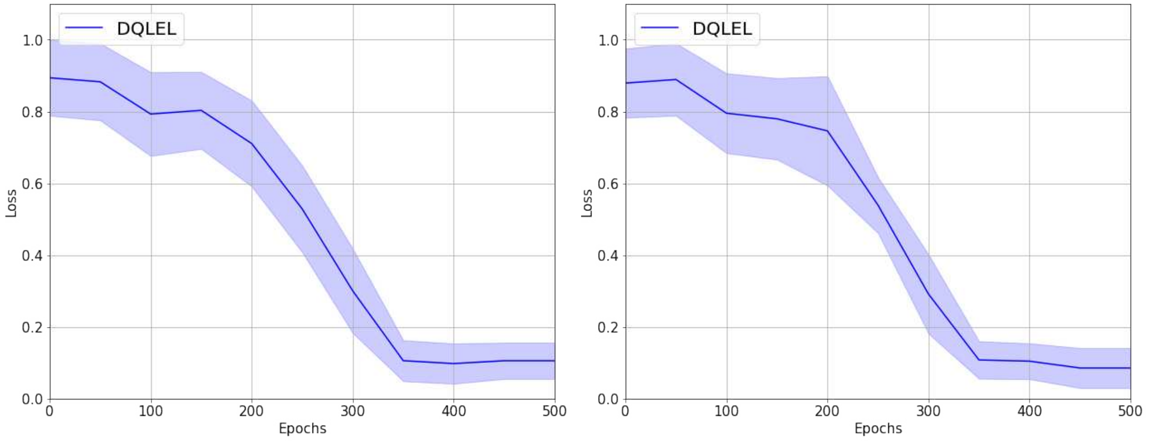
and others as NLoS at random. More specifically, after creating an environment with random obstacles, the mobile user's beginning position is chosen at random. We use randomly generated environments to construct a robust agent that can perform effectively in new settings. To be more realistic and to deal with a multi-user indoor environment, we consider several moving obstacles in addition to the fixed obstacles in the this case study. Following Reference [198], we use an increasingly complex indoor environment to simulate a multi-user scenario. In this case study, random obstacles are considered throughout the environment as the mobile devices. As the first step towards development of a fully autonomous agent for UWB-based indoor localization, we considered a DRL model with one agent that should be localized in the environment, and other mobile devices are treated as mobile obstacles. We, therefore, assumed N_o mobile devices (mobile obstacles) in our testbed, with at least one mobile obstacle in each sub-area in addition to the agent. It is assumed that mobile obstacles randomly move through the environment, where their location will be changed in each epoch. Fig. 5.28 illustrates that although a multi-user environment takes much more time to learn the optimal policy, the loss function converges, illustrating that the model is well-trained.

Furthermore, we consider a numerical reward function, where the reward function is based on the link condition instead of the location error, as follows

$$\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} 10, & \textbf{C1.}\ \mathcal{MD}_t \leq \mathcal{MD}_{th},\ c_{i,t} = 1, c_{j,t} = 1, \\ 5, & \textbf{C2.}\ \mathcal{MD}_t > \mathcal{MD}_{th},\ c_{i,t} = 1, c_{j,t} = 1, \\ -5, & \textbf{C3.}\ \mathcal{MD}_t \leq \mathcal{MD}_{th},\ c_{i,t} = 0, c_{j,t} \in \{0,1\}, \\ -10, & \textbf{C4.}\ \mathcal{MD}_t > \mathcal{MD}_{th},\ c_{i,t} = 0, c_{j,t} \in \{0,1\}. \end{cases}, \tag{197}$$

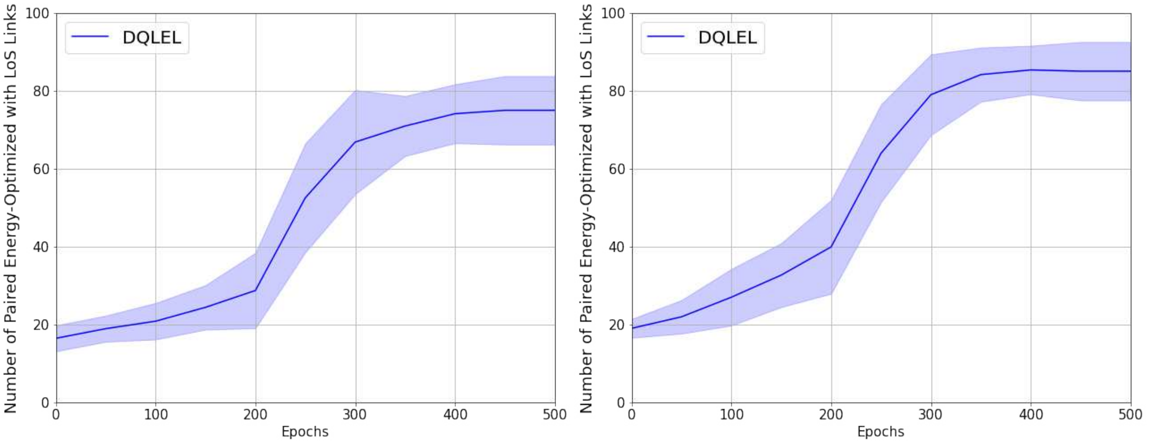
where $c_{i,t}$ and $c_{j,t}$ indicate the links' condition associated with the given action $\boldsymbol{a}$. In DQLEL, we define a $(N_u \times N_l)$ link condition matrix, denoted by $\mathbf{C}$, where the t^{th} column illustrates the link condition established between the user at point (x_t, y_t) and all UWB beacons. Then, $c_{i,t} = 1$ if the link between the user at location (x_t, y_t) and the i^{th} UWB beacon is LoS, otherwise $c_{i,t} = 0$. According to Eq. (197), both location accuracy and the balanced energy consumption of beacons are considered as the reward function, where the former is stated in terms of the link condition $c_{i,t}$ and $c_{j,t}$ and the latter one is expressed as $\mathcal{MD}_t$. More precisely, the reward function in Eq. (197) illustrates the following four connection types for the selected pair of UWB beacons: (**C1.**) Energy-optimized with LoS links; (**C2.**) Non energy-optimized with LoS links; (**C3.**) Energy-optimized with NLoS links, and; (**C4.**) Non energy-optimized with NLoS links. An energy-optimized link is referred to the action where the battery life deviation of all UWBs is less than a pre-determined threshold $\mathcal{MD}_{th}$; otherwise, it is called a non energy-optimized connection. Moreover, $c_{i,t} = 0$, $c_{j,t} \in \{0,1\}$ means that at least one link of the selected pair of UWB beacons is NLoS, leading to a remarkable location error. Therefore, to achieve a high location accuracy, both UWB beacons should be LoS, i.e., $c_{i,t} = 1$, $c_{j,t} = 1$. Note that, in the worst-case scenario, the selected pair are non energy-optimized with NLoS links. Our goal is to increase the energy-optimized with LoS connection type, while reducing the non energy-optimized with NLoS connections.

Convergence of the proposed DQLEL framework with the numerical reward function is evaluated in Fig. 5.29. According to Eq. (191), the learning process is performed by minimizing the loss function, which is the mean-squared error of the target optimal Q-function with the minibatch updates. Fig. 5.29 illustrates that the proposed DQLEL framework converges after 350 epochs. Moreover,



(a) (b)

Figure 5.29: The convergence of the proposed DQLEL framework for (a) $N_u = 4$, and (b) $N_u = 6$.

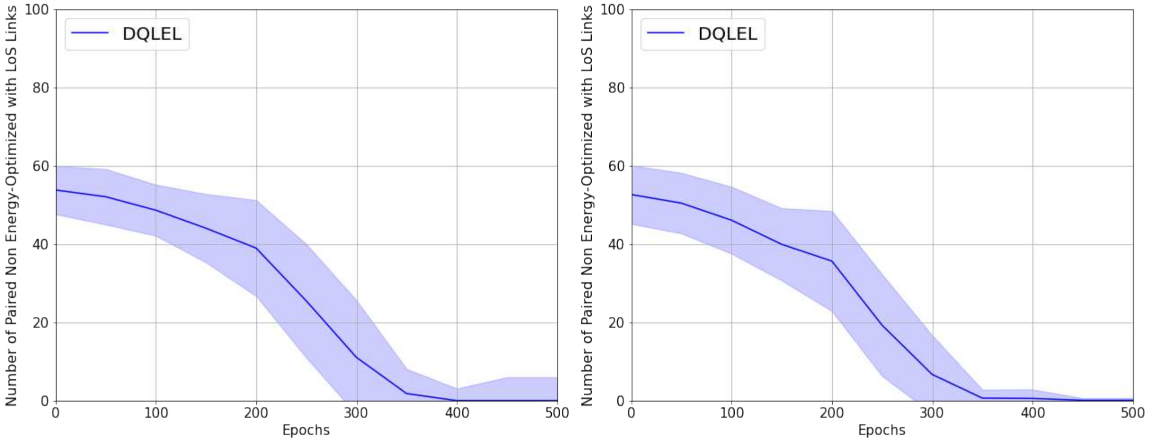


(a) $N_u = 4$ (b) $N_u = 6$

Figure 5.30: Number of paired energy-optimized with LoS Links (a) $N_u = 4$, and (b) $N_u = 6$.

Figs. 5.30-5.33 illustrate the number of connections of each type in different epochs. We also investigate the effect of the number of UWB beacons in each sub-area in Figs. 5.30-5.33.

Fig. 5.30 illustrates the number of paired energy-optimized with LoS links in the proposed DQLEL framework for $N_u = 4$ and $N_u = 6$. It is worth mentioning that given the limited transmission range of the UWB beacons, it is common to have a small number of beacons in each sub-area, typically, 4 beacons are considered. To evaluate the effects the number of beacons on the DQLEL framework, we consider the common scenario with 4 beacons together with a second scenario with 6 UWB beacons. The environment for both scenarios is



(a)

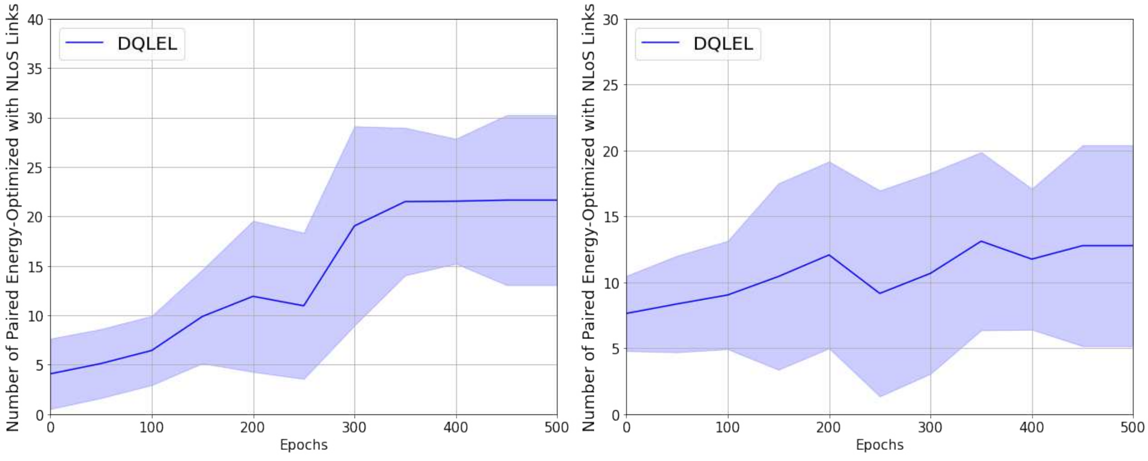
(b)

Figure 5.31: Number of paired non energy-optimized with LoS links for (*a*) $N_u = 4$, and (*b*) $N_u = 6$.

considered to be similar, therefore, the number of NLoS links in both cases would be the same, while the number of possible actions in $N_u = 4$ and $N_u = 6$ are 6 and 15, respectively. Under the assumption that in each location, at least one link is NLoS, then the probability of establishing LoS connections for $N_u = 4$ and $N_u = 6$ are 50% and 66%, respectively. Intuitively speaking, this means that when the action space is small (e.g., 6 actions when there are 4 beacons compared to 15 actions when we have two extra beacons), it is less likely to have paired LoS links. According to the results in Fig. 5.30, it can be observed that the number of energy-optimized with LoS connections increases as the number of epochs grows and converges after about 330 epochs to 75% and 85% for $N_u = 4$ and $N_u = 6$, respectively.

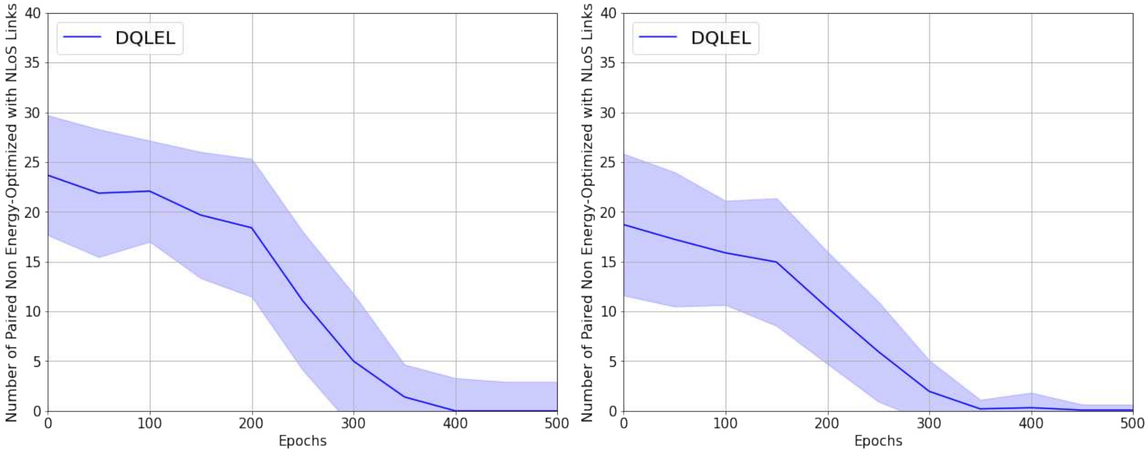
Fig. 5.31 illustrates the number of paired LoS links that make the deviation of remaining battery life of UWB beacons become greater than $\mathcal{MD}_{th}$. As it can be seen from Figs. 5.30 and 5.31, most LoS links at the earlier epochs are non energy-optimized. By increasing the epochs, however, the number of non energy-optimized LoS links decreases resulting in more energy-optimized LoS connections. According to the results in Fig. 5.31, the number of non energy-optimized links converges to zero after about 350 epochs in both $N_u = 4$ and $N_u = 6$.

The number of energy-optimized and non energy-optimized NLoS connections are depicted in Figs. 5.32 and 5.33, respectively. By a similar argument, non



(a) $N_u = 4$ (b) $N_u = 6$

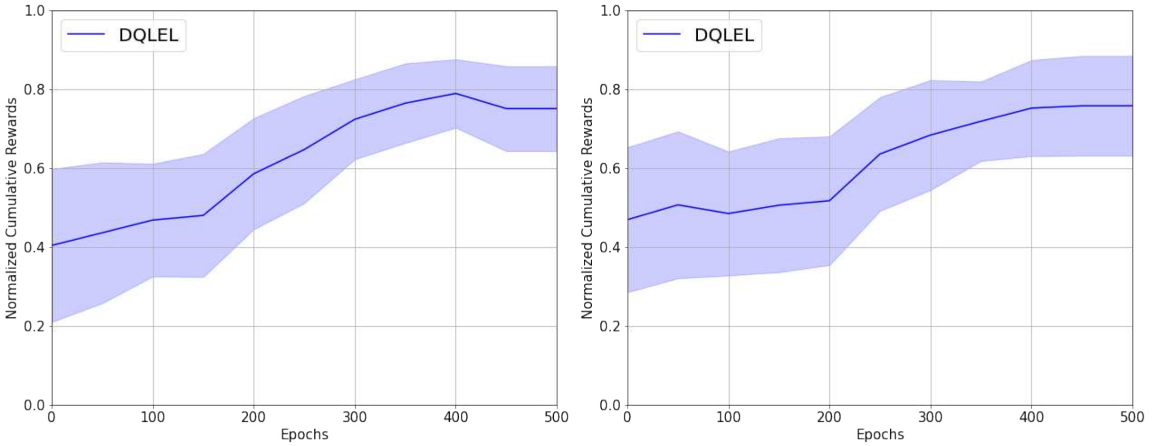
Figure 5.32: Number of paired energy-optimized with NLoS links for (a) $N_u = 4$, and (b) $N_u = 6$.



(a) $N_u = 4$ (b) $N_u = 6$

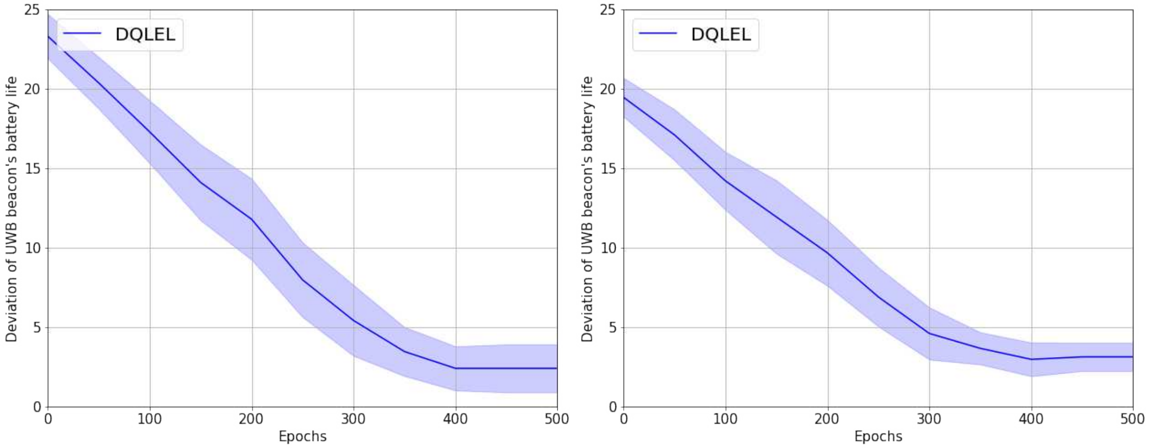
Figure 5.33: Number of paired non energy-optimized with NLoS links for (a) $N_u = 4$, and (b) $N_u = 6$.

energy-optimized NLoS links in Fig. 5.33 experience a remarkable reduction, eventually converging to zero after 350 epochs, which results in a slight increase in the energy-optimized NLoS connections. By considering the fact that the proposed DQLEL framework needs to maintain a trade-off between two objectives, i.e., the link condition and energy consumption of UWB beacons, the small growth in number of energy-optimized NLoS connections in Fig. 5.32 is acceptable. Fig. 5.34 illustrates the normalized cumulative rewards of the agent in each epoch. According to the reward definition in Eq. (190), energy-optimized LoS connections result in a considerable increase in the cumulative rewards. According to the results in Fig. 5.34, increasing the number of epochs increases the cumulative rewards, showing that the model is well-trained.



(a) $N_u = 4$ (b) $N_u = 6$

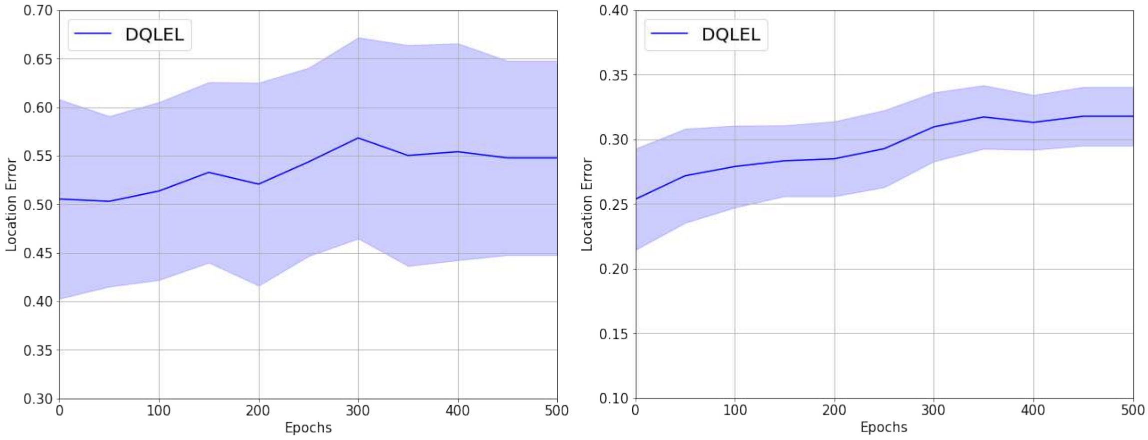
Figure 5.34: The variation of normalized cumulative rewards versus different epochs for (a) $N_u = 4$, and (b) $N_u = 6$.



(a) $N_u = 4$ (b) $N_u = 6$

Figure 5.35: Deviation of UWB beacons battery life versus different epochs for (a) $N_u = 4$, and (b) $N_u = 6$.

Fig. 5.35 evaluates the deviation of the UWB beacons' battery life, obtained by Eq. (189), versus different epochs for (a) $N_u = 4$, and (b) $N_u = 6$. Note that, larger value of batteries' deviation indicates that certain UWB beacons are involved in localization more than others. In this model, we assume that each epoch is terminated if the current time reaches a pre-determined time value, or the battery of at least one UWB beacon is completely drained before the time threshold. Without considering the UWB beacon's energy consumption, the agent identifies a pair of UWB beacons with LoS links, draining the battery of those corresponding beacons by being repeatedly selected. The main goal of the proposed DQLEL framework is to select UWB beacons with LoS links while minimizing the deviation of UWB beacons' battery lives. As it can be seen from Fig. 5.35, the deviation of UWB beacons' battery lives decreases as the number of epochs grows and converges after about 350 epochs to 2.4 and 3.8 for $N_u = 4$ and $N_u = 6$, respectively. In addition, Fig. 5.36 illustrates the location error versus different epochs, which is obtained by Eq. (188). As it can be seen from Fig. 5.36, the location error slightly increases by increasing the number of epochs, which is negligible. The main reason behind this is that there is a trade-off between the number of LoS links (accuracy) and the deviation of UWB beacons battery life. This completes the evaluation of the DQLEL framework.



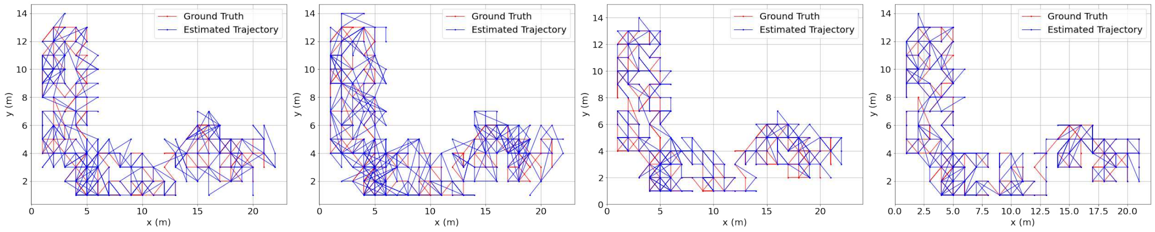
(a) $N_u = 4$ (b) $N_u = 6$

Figure 5.36: Location error (m) versus different epochs for (a) $N_u = 4$, and (b) $N_u = 6$.

Performance Comparisons

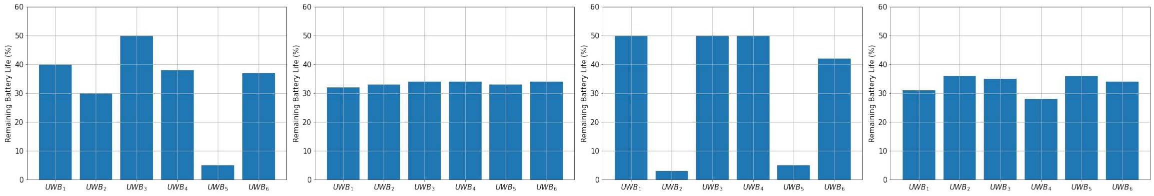
To the best of our knowledge, there is no RL-based UWB node selection framework that studied both LoS/NLoS and energy perspectives, for comparison purposes. Therefore, we introduce three baseline models for comparisons:

- ***Non Energy-Optimized DRL (NE-DRL) LoS/NLoS UWB Node***



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

Figure 5.37: Ground truth and estimated random trajectories by using (a) NN-NS, (b) RNS, (c) NE-DRL, and (d) DQLEL frameworks.



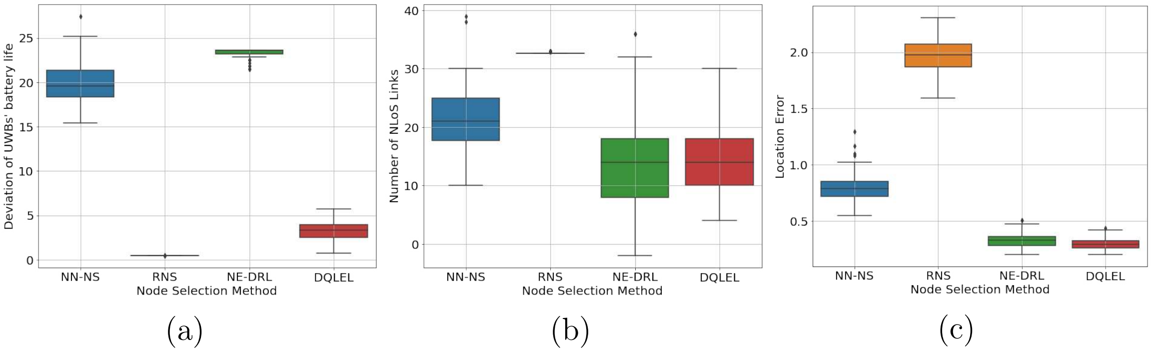
(a) (b) (c) (c)

Figure 5.38: The percentage of remaining battery life of UWB beacons in one sub-area after 50 iterations by using (a) NN-NS, (b) RNS, (c) NE-DRL, and (d) DQLEL frameworks.

Selection: Similar to the proposed DQLEL framework, i.e., all the parameters are the same, with the difference that in this baseline model, we just consider the link condition as the reward function. For an action that both UWB links are LoS connections, the reward r_t is set equal to 1, otherwise, it equals -1.

- ***Random UWB Node Selection (RNS):*** In this framework, a pair of UWB beacons are randomly selected for localization without considering the remaining battery life of UWB beacons and channel conditions.
- ***Nearest Neighbor UWB Node Selection (NN-NS):*** Similar to the previous one, without considering the remaining battery life of UWB beacons and channel conditions, two nearest UWB beacons are selected for localization.

For comparison purposes, we consider a (24×15) m^2 rectangular indoor environment. Fig. 5.37 illustrates the estimated random trajectory of the mobile user by using NN-NS, RNS, NE-DRL, and DQLEL frameworks. Since the mobile user in the NE-DRL and DQLEL frameworks is trained to be localized by LoS beacons, the estimated trajectory is almost the same as the ground truth (see Fig. 5.37). Similarly, Fig. 5.38 compares the remaining battery life of UWB beacons after 50 iterations. According to the results in Fig. 5.38, the battery of certain UWB beacons are completely drained without applying the DQLEL framework, while the remaining battery of UWB beacons are almost the same in the DQLEL framework, leading to a remarkable increase in the life time of the infrastructure.



(a) (b) (c)

Figure 5.39: (a) A comparison of different node selection methods based on (a) Deviation of UWBs' battery life; (b) Number of NLoS links; (c) Location error.

It should be noted that although the mobile user uniformly moves across the environment in 8 directions given a starting location, the initial user's location as well as the user's mobility pattern are changed in each epoch. Fig. 5.38 in the revised manuscript is included to illustrate random (non-uniform) nature of the user's trajectory in the experimental testbed. According to the user's random trajectory shown in Fig. 5.38, it is evident that all locations are not being explored equally. Moreover, regarding the popularity of some specific spots, we will have the following two scenarios: (i) *Popular sub-area*: It should be noted that the main goal of the DQLEL framework is to maintain a balance between the remaining battery life of UWB beacons located in *one sub-area*. Due to the limited transmission range of UWB beacons, only beacons in a specific sub-area are involved for localizing the mobile user. Therefore, when a specific sub-area in the experimental indoor environment is more popular than other areas, it is expected that the corresponding beacons' batteries drain sooner than others. To address this issue practically, such sub-areas should be identified and treated differently and possibly non-battery powered beacons should be utilized instead. (ii) *Popular location in one sub-area*: In this case, the mobile user

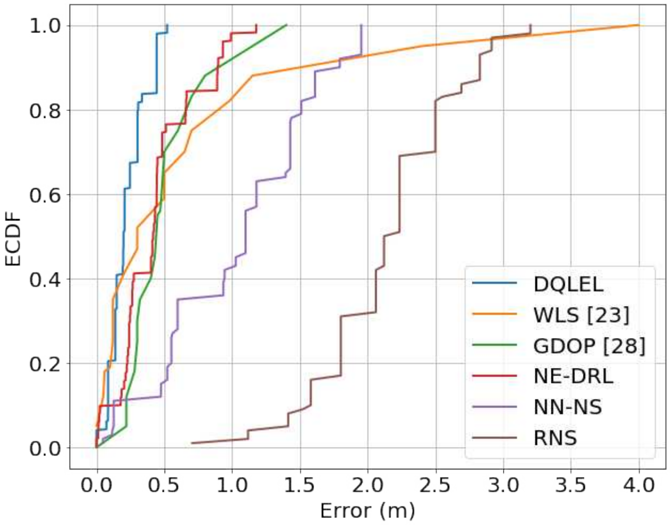


Figure 5.40: A comparison with state-of-the-arts based on the location error ECDF.

visits a given spot in one sub-area more frequently. To illustrate the superiority of the proposed DQLEL framework in handling such scenarios, we compare DQLEL's performance with the Non Energy-Optimized DRL (NE-DRL) LoS/NLoS UWB node selection, where there is no consideration for preserving a balance between beacons' energy consumption. According to the results shown in Fig. 5.39(c) and Fig. 5.39(d), in comparison to the NE-DRL framework that the learnt behavior drains out batteries in popular spots, the remaining battery of beacons will remain almost the same in the DQLEL framework because of our multi-objective reward function. We can, therefore, state that in sub-areas with specific popular spots, the DQLEL framework efficiently attempts to learn the policy that minimizes the deviation of UWBs' battery lives.

Fig. 5.39(a) compares the performance of the proposed DQLEL framework with NE-DRL, RNS, and NN-NS schemes in terms of the deviation of UWBs' battery life. Due to the random movement of the mobile user and the random UWB node selection approach of the RNS framework, the remaining battery life of the UWB beacons are almost the same. Therefore, the mean deviation of UWBs' battery life is almost zero. In other words, the RNS approach performs reasonably well in terms of energy distribution, however, fails to accurately localize the target. In contrary to the RNS method, the mean deviation of UWBs' battery life of the NN-NS and NE-DRL is considerably high. The main reason behind this high mean deviation of UWBs' battery life is that the NN-NS and NE-DRL frameworks select the nearest beacons and the beacons with LoS links for localization, respectively, without taking into account energy considerations. As a result, the battery of certain UWB beacons are completely drained leading to a high deviation of UWBs' battery life. The proposed DQLEL framework, however, provides a low battery life deviation, while enhancing the location accuracy by choosing UWB beacons with LoS links. Fig. 5.39(b) illustrates the number of NLoS links established by different node selection methods. Note that, using less number of NLoS links provides the higher location accuracy. As it can be seen from Fig. 5.39(b), the proposed DQLEL framework and NE-DRL experience lower number of NLoS links. RNS method, however, results in the highest number of NLoS connections since there is no link condition criteria. By considering the fact that the nearest UWB beacons provide LoS links with a higher probability, the possibility of selecting UWB beacons with NLoS links in the NN-NS scheme is lower than the RNS method. Fig. 5.39(c) compares the performance of the proposed DQLEL framework with NE-DRL, RNS, and NN-NS schemes from the aspect of the location error. By a similar argument, which is used for the number of NLoS links, localization by applying RNS method leads to the highest location error. The proposed DQLEL and NE-DRL frameworks have the lowest position error since they both select LoS connections. Finally, to demonstrate the superiority of the proposed DQLEL framework in comparison to its state-of-the-art counterparts, we calculate the Empirical Cumulative Distribution Function (ECDF) of the WLS [101], GDOP [103], NE-DRL (which itself is a simplified version of the DQLEL framework), RNS, and NN-NS anchor selection schemes. According to the results shown in Fig. 5.40, the location error of the DQLEL framework is lower than other methods.

5.5.4 Reasonability of the Reward Function

Intuitively speaking, our goal is to keep a trade-off between the remaining battery life of beacons and localization accuracy, i.e., jointly minimizing location error and the battery life deviation. For this reason, we considered a multi-objective DRL model, where multiplicative scalarization function [199] is used to convert the problem at hand into its single objective counterpart. To analytically illustrate the reasonability of the reward function, it is required to show that if the agent takes an action (i.e., a set of UWB beacons) resulting in lower battery life deviation of beacons and location error, this should result in higher reward value by using the reward function in Eq. (190). Without loss of generality, we consider a scenario where the user has two possible actions at time t to follow (denoted by a_1 and a_2) each of which consisting of a set of selected UWB beacons. In such a scenario, the following possible cases can happen:

Case 1 ($\mathcal{MD}_{t,1} < \mathcal{MD}_{t,2} \leq \mathcal{MD}_{th}$ and $\mathcal{ER}_{t,1} < \mathcal{ER}_{t,2} \leq \mathcal{E}_{th}$): Assume that the location error and the battery life deviation of action set a_1 is lower than that of a_2 (i.e., $\mathcal{MD}_{t,1} < \mathcal{MD}_{t,2}$ and $\mathcal{ER}_{t,1} < \mathcal{ER}_{t,2}$). The following shows that in such a scenario as expected (wanted) the reward value of action a_1, denoted by $\mathcal{R}_1$, is greater than $\mathcal{R}_2$, associated with action a_2. In such scenario, the reward value associate with both actions is defined as $\dfrac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}$ and $\dfrac{1}{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= \frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}} - \frac{1}{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}} \\ &= \frac{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2} - \mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}} > 0. \qquad (198)\end{aligned}$$

Since $\mathcal{MD}_{t,1} < \mathcal{MD}_{t,2}$ and $\mathcal{ER}_{t,1} < \mathcal{ER}_{t,2}$, therefore, we can see that term $\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2})$ is positive, which means an action with lower location error and battery life deviation of beacons results in higher reward value.

Case 2 ($\mathcal{MD}_{th} < \mathcal{MD}_{t,1} < \mathcal{MD}_{t,2}$ and/or $\mathcal{ER}_{th} < \mathcal{ER}_{t,1} < \mathcal{ER}_{t,2}$): In this case, the reward value associate with both actions is defined as $-\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}$ and $-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= -\mathcal{MD}_{t,1}\mathcal{ER}_{t,1} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= -\mathcal{MD}_{t,1}\mathcal{E}_{t,1} + \mathcal{MD}_{t,2}\mathcal{E}_{t,2} > 0. \qquad (199)\end{aligned}$$

Therefore, we showed that in this case, the reward of action a_1 is greater than action a_2, which means the lower location error and battery life deviation of beacons results in higher reward value.

Case 3 ($\mathcal{MD}_{t,1} < \mathcal{MD}_{t,2} \leq \mathcal{MD}_{th}$ and $\mathcal{ER}_{t,1} \leq \mathcal{ER}_{th} < \mathcal{ER}_{t,2}$): In this case, the reward value of action a_1 is calculated by $\dfrac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}$, while $\mathcal{R}(s_t, a_{t,2}) =$

$-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, so we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= \frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= \frac{1 + \mathcal{MD}_{t,2}\mathcal{E}_{t,2}\mathcal{MD}_{t,1}\mathcal{E}_{t,1}}{\mathcal{MD}_{t,1}\mathcal{E}_{t,1}} > 0, \end{aligned} \tag{200}$$

which is always positive confirming the expected result.

Case 4 ($\mathcal{MD}_{th} < \mathcal{MD}_{t,1} < \mathcal{MD}_{t,2}$ and $\mathcal{ER}_{t,1} < \mathcal{ER}_{th} < \mathcal{ER}_{t,2}$): In this case, the reward value associate with both actions is defined as $-\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}$ and $-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= -\mathcal{MD}_{t,1}\mathcal{ER}_{t,1} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= -\mathcal{MD}_{t,1}\mathcal{E}_{t,1} + \mathcal{MD}_{t,2}\mathcal{E}_{t,2} > 0. \end{aligned} \tag{201}$$

which is always positive, again confirming that lower location error and battery life deviation of beacons results in higher reward value.

Case 5 ($\mathcal{MD}_{t,1} \leq \mathcal{MD}_{th} < \mathcal{MD}_{t,2}$ and $\mathcal{ER}_{t,1} \leq \mathcal{ER}_{t,2} < \mathcal{ER}_{th}$): Similar to Case 3, the reward value of action a_1 is calculated by $\frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}$, while $\mathcal{R}(s_t, a_{t,2}) = -\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, so we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= \frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= \frac{1 + \mathcal{MD}_{t,2}\mathcal{E}_{t,2}\mathcal{MD}_{t,1}\mathcal{E}_{t,1}}{\mathcal{MD}_{t,1}\mathcal{E}_{t,1}} > 0. \end{aligned} \tag{202}$$

Case 6 ($\mathcal{MD}_{t,1} \leq \mathcal{MD}_{th} < \mathcal{MD}_{t,2}$ and $\mathcal{ER}_{th} < \mathcal{ER}_{t,1} < \mathcal{ER}_{t,2}$): Similar to Case 4, the reward value associate with both actions is defined as $-\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}$ and $-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= -\mathcal{MD}_{t,1}\mathcal{ER}_{t,1} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= -\mathcal{MD}_{t,1}\mathcal{E}_{t,1} + \mathcal{MD}_{t,2}\mathcal{E}_{t,2} > 0. \end{aligned} \tag{203}$$

Case 7 ($\mathcal{MD}_{t,2} < \mathcal{MD}_{t,1} \leq \mathcal{MD}_{th}$ and $\mathcal{E}_{t,1} < \mathcal{E}_{t,2} \leq \mathcal{E}_{th}$): In such scenarios that an action cannot satisfy both metrics (lower location error and lower battery deviation), a trade-off between these metrics will be achieved by this reward function. For instance, in such a case that $\mathcal{E}_{t,1} < \mathcal{E}_{t,2} \leq \mathcal{E}_{th}$, while $\mathcal{MD}_{t,2} < \mathcal{MD}_{t,1} \leq \mathcal{MD}_{th}$, the reward value associate with both actions is defined as $\frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}$ and $\frac{1}{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= \frac{1}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}} - \frac{1}{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}} \\ &= \frac{\mathcal{MD}_{t,2}\mathcal{ER}_{t,2} - \mathcal{MD}_{t,1}\mathcal{ER}_{t,1}}{\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}},\end{aligned} \tag{204}$$

where the reward value associated with action a_1 would be greater than a_2 (i.e., $\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) > 0$) if $\mathcal{MD}_{t,1} < \frac{\mathcal{E}_{t,1}}{\mathcal{E}_{t,2}}\mathcal{MD}_{t,2}$, otherwise, action a_2 results in higher reward.

Case 8 ($\mathcal{MD}_{th} < \mathcal{MD}_{t,2} < \mathcal{MD}_{t,1}$ and $\mathcal{E}_{t,1} < \mathcal{E}_{t,2} < \mathcal{E}_{th}$): In this case, the reward value associate with both actions is defined as $-\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}$ and $-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$, therefore, we have

$$\begin{aligned}\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) &= -\mathcal{MD}_{t,1}\mathcal{ER}_{t,1} - (-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}) \\ &= -\mathcal{MD}_{t,1}\mathcal{E}_{t,1} + \mathcal{MD}_{t,2}\mathcal{E}_{t,2} > 0.\end{aligned} \tag{205}$$

where the reward value associated with action a_1 would be greater than a_2 (i.e., $\mathcal{R}(s_t, a_{t,1}) - \mathcal{R}(s_t, a_{t,2}) > 0$) if $\mathcal{MD}_{t,1} < \frac{\mathcal{E}_{t,2}}{\mathcal{E}_{t,1}}\mathcal{MD}_{t,2}$, otherwise, action a_2 results in higher reward. Other cases, such as case $\mathcal{MD}_{th} < \mathcal{MD}_{t,2} < \mathcal{MD}_{t,1}$ and $\mathcal{E}_{th} < \mathcal{E}_{t,1} < \mathcal{E}_{t,2}$), case $\mathcal{MD}_{t,2} < \mathcal{MD}_{t,1} < \mathcal{MD}_{th}$ and $\mathcal{E}_{th} < \mathcal{E}_{t,1} < \mathcal{E}_{t,2}$, case $\mathcal{MD}_{th} < \mathcal{MD}_{t,2} < \mathcal{MD}_{t,1}$ and $\mathcal{E}_{t,1} < \mathcal{E}_{th} < \mathcal{E}_{t,2}$, case $\mathcal{MD}_{t,2} < \mathcal{MD}_{th} < \mathcal{MD}_{t,1}$ and $\mathcal{E}_{t,1} < \mathcal{E}_{th} < \mathcal{E}_{t,2}$, and case $\mathcal{MD}_{t,2} < \mathcal{MD}_{th} < \mathcal{MD}_{t,1}$ and $\mathcal{E}_{th} < \mathcal{E}_{t,1} < \mathcal{E}_{t,2}$, where the reward value associate with both actions is defined as $-\mathcal{MD}_{t,1}\mathcal{ER}_{t,1}$ and $-\mathcal{MD}_{t,2}\mathcal{ER}_{t,2}$ are exactly the same as Case 8.

5.6 Accelerating Learning Process of RL Framework

In our previous work [13], the mobile user is autonomously trained via an RL model to be localized by a set of UWB beacons with LoS connections. The main objective of the RL model within the indoor localization domain is to learn an optimal or near-optimal policy that maximizes the location accuracy. One of the most important challenges of existing RL models [13], however, is that the optimal policy should be learned by the interaction of the agent (mobile user) with the environment (i.e., via trial and error), without any prior information, especially when the model is just initialized. Consequently, it may take a long time for the RL model to reach the optimal policy. Another challenge is the generalization ability of the pre-trained RL model to be used in a new and different environment, where the density/location of obstacles is changing over the time/environment. To tackle these issues, Uchendu *et al.* [202] proposed the Jump-Start RL (JSRL) model, where the agent use a guide-policy instead of a random one at the beginning of the learning process. Consequently, the learning process is accelerated and the RL generalization ability is highly improved.

Motivated by the above discussion, we introduce the Jump-start RL-based Uwb NOde selection (JUNO) framework with the application to indoor localization. The main novelty of this work is the design of an autonomous and real-time anchor node selection, where the key objective is to accelerate the location accuracy improvement. Towards this goal, a combination of the guide and exploration policies is used, where the guide-policy significantly speeds up the early learning phase of the RL model to converge to the optimal location accuracy. Furthermore, since any random guide-policy can be used in the JUNO framework, the generalization ability of the pre-trained RL frameworks improves. Simulation results illustrate that the proposed JUNO framework outperforms its state-of-the-art counterparts in terms of the cumulative rewards and location error even in an ultra-dense indoor environment. In following, we first introduce the JSRL model, and then present details of the proposed JUNO framework.

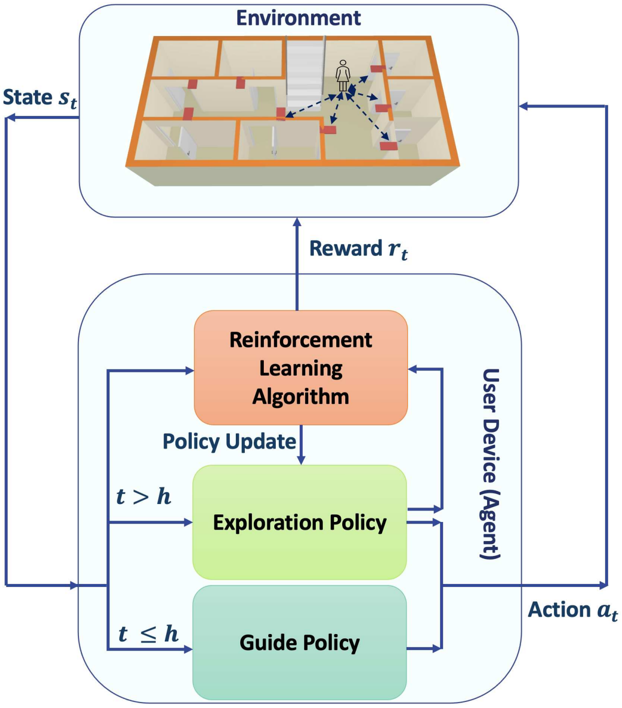


Figure 5.41: The block diagram of the proposed JUNO anchor selection framework.

5.6.1 JSRL model

Conventionally, an RL-based agent selects a random action a_t at the beginning of the learning process, where there is no prior information resulting in a long time to reach the optimal policy π. The main difference between the conventional RL and JSRL model [202] is that the agent in the JSRL framework has access to two policies, called guide-policy $\pi^g(a|s)$, and the exploration policy $\pi^e(a|s)$. While the exploration policy is the same policy used in conventional RL models, which will be updated during the training, the guide-policy is a fixed prior policy, learned via an RL model or manually/randomly constructed. Since the Q-table is initialized with zeros, the agent of the JSRL model follows $\pi^g(a|s)$ instead of randomly selecting an action a_t based on the untrained policy $\pi^e(a|s)$. Considering the fact that the distribution of the data under policy $\pi^g(a|s)$ is not exactly the same as the policy $\pi^e(a|s)$, the state space of the policy $\pi^e(a|s)$ would be different. Therefore, it is essential to gradually transit the data collection from the policy $\pi^g(a|s)$ to policy $\pi^e(a|s)$. Consequently, given an RL model with horizon H, we define guide-step $h \leq H$ as the number of steps where the agent uses policy $\pi^g(a|s)$, initialized with H and gradually decreases over the course of training. More precisely, the action is selected based on $\pi^g(a|s)$ for h steps at the initial stage of each training episode, continuing with $\pi^e(a|s)$ for remaining $(H-h)$ steps.

5.6.2 JUNO Anchor Selection

Following Reference [202], we use the JSRL model in our proposed JUNO framework to accelerate the learning process. Due to the dynamic nature of indoor venues, caused by varying environmental conditions, the proposed JUNO framework seeks to train the mobile user to autonomously find the optimal LoS connections at any given time and place. The proposed JUNO framework consists of the following main components:

(i) Agent: In the proposed JUNO framework, the mobile user operates as the agent, interacting with the environment via a set of actions.

(ii) State-Space: The state $\mathbf{s}_t \in \mathcal{S}$ is defined as the user's location (x_t, y_t) at time slot t. Following Reference [206], we discretize the indoor environment into $N_l = N_x \times N_y$ points, where x_t and y_t are obtained as

$$\begin{aligned} x_t &= x_{t-1} + \zeta_x, \qquad 0 \leq x_t \leq N_x && (206) \\ \text{and } y_t &= y_{t-1} + \zeta_y, \qquad 0 \leq y_t \leq N_y. && (207) \end{aligned}$$

where $\zeta_x, \zeta_y \in \{-1, 0, 1\}$ are random numbers, indicating the user's movement in x-axis and y-axis [206], respectively. It should be noted that although RL models with a continuous and high-dimensional state-space [204] provide higher resolution and precise localization, they suffer from high complexity, making them inefficient for real-time applications requiring low latencies. The benefits of the discrete RL models in the context of indoor localization [203, 206], therefore, make us to choose discrete RL over its continuous counterpart.

(iii) Action-Space: The action space is defined as a set of nearby UWB beacons, where the user's location can be determined by extracting the time information from the received signal of the corresponding beacons. The cardinality of the action space, denote by N_a, is given by

$$N_a = \frac{N_u!}{(N_u - N_r)!\ N_r!}, \tag{208}$$

where N_r represents the number of required beacons for localization, depending on some parameters, such as the indoor localization framework (i.e., ToA, TDoA, and Two Way Ranging (TWR)), and the dimension of the experimental environment, i.e., 2D or 3D area. Considering the fact that at least two UWB beacons are required for the TDoA-based localization scheme in a 2D environment, the selected action is a vector, denoted by $\mathbf{a} = [a_i, a_j]$, where a_i, a_j represent UWB_i and UWB_j, respectively.

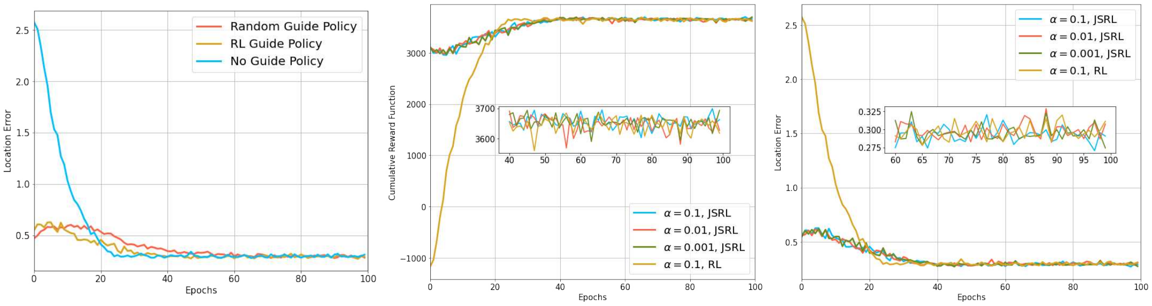
(iv) Reward: As stated previously, the main objective of the proposed JUNO framework is to minimize the location error caused by UWB beacons with NLoS connections. Therefore, after taking action $\mathbf{a}_t$, the estimated user's location $(x_{es,t}^{(i,j)}, y_{es,t}^{(i,j)})$ is calculated, where the superscript (i,j) indicates that the estimated location is obtained by the received signals of UWB_i and UWB_j. Considering the fact that even if one of these two connections is NLoS, we will face with a remarkable location error, the combination of UWB_i and UWB_j at any location/time is of paramount importance. For this reason, the reward function $\mathcal{R}(s_t, a_t)$ is defined as

$$\mathcal{R}(s_t, a_t) = \begin{cases} \dfrac{1}{\mathcal{E}_t}, & \mathcal{E}_t \leq \mathcal{E}_{th}, \\ -\mathcal{E}_t, & \text{o.w.} \end{cases}, \tag{209}$$

where $\mathcal{E}_{th}$ is a pre-defined threshold value for the maximum acceptable location error [203], and $\mathcal{E}_t$ denotes the location error at time slot t, calculated as

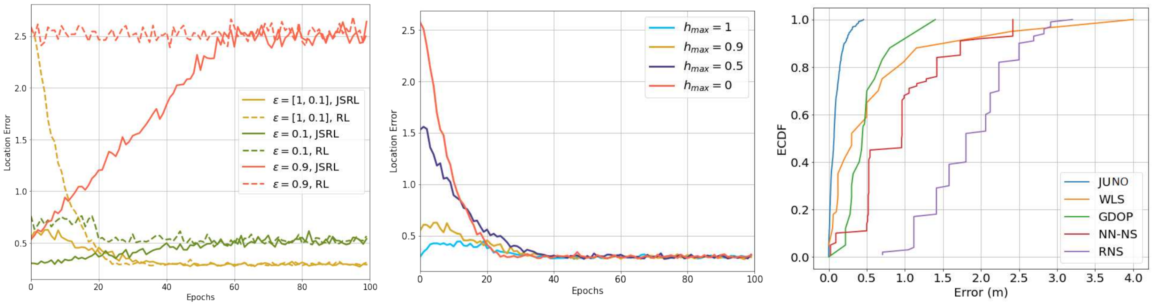
$$\mathcal{E}_t = \sqrt{(x_t - x_{es,t}^{(i,j)})^2 + (y_t - y_{es,t}^{(i,j)})^2}. \tag{210}$$

This completes presentation of the proposed JUMP anchor selection framework, next, we will describe our testbed and simulation results.



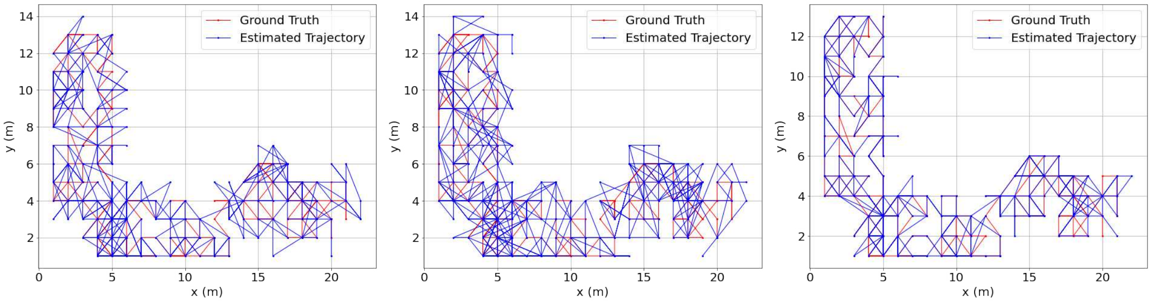
(a) (b) (c)

Figure 5.42: Investigating the effect of (a) Guide-policy on the location error; (b) Learning rate α on the cumulative rewards, and; (c) Learning rate α on the location error.



(a) (b) (c)

Figure 5.43: Investigating the effect of (a) ϵ on the location error; (b) h_{max} on the location error, and; (c) ECDF on the location error.



(a) (b) (c)

Figure 5.44: Ground truth and estimated random trajectories by using: (a) Nearest neighbor; (b) Random, and; (c) JUNO frameworks.

5.6.3 Simulation Results

To evaluate the performance of the proposed JUNO framework, we consider an experimental indoor area such as an office building with the size of $(60 \times 50)\ m^2$, which is compromised of several non-overlapping sub-areas [106,206]. Following

Reference [206], each sub-area is discretized into several square zones, where the dimension of each zone is $(1 \times 1)\ m^2$. Although the location resolution, i.e., the number of discretized points in the indoor environment N_l, is proportional to the location accuracy, it also results in higher state-space, complexity, and the respond-time. Therefore, there should be a trade-off between the location resolution and the respond-time of the learning model. Despite the recent RL-based localization works [203,206], where the environment is divided into a grid of $(5 \times 5)\ m^2$ and $(3 \times 3)\ m^2$ cells, respectively, we assume higher resolution of $(1 \times 1)\ m^2$ to improve the location accuracy. It is assumed that there is one mobile user, randomly moving through the network in 8 directions based on Eqs. (206) and (207), where the mobile user is placed at zones' center [206]. At each location, the transmitted signal by the mobile user are received by a set of nearby UWB beacons. Due to the obstacles in the environment, the received signal would be LoS or NLoS connections, where it is assumed that the channel condition of UWB_i, for $(1 \leq i \leq N_u)$, is determined randomly at each zone to initialize the environment.

Fig. 5.42(a) illustrates the effect of the guide-policy on the proposed JUNO framework. As shown in Fig. 5.42(a), using a random Q-table or the one obtained by an RL model as the guide-policy outperforms the conventional RL approach, accelerating the learning process of the JUNO framework, improving the location accuracy, and increasing the RL's generalization capabilities. We also investigate the effect of learning rate α on the proposed JUNO framework to obtain the best value of α. As shown in Figs. 5.42(b)-(c), increasing the number of epochs decreases the location error and increases the cumulative rewards, illustrating that the model is well-trained. Moreover, it is evident that the learning rate has not a great impact on the location accuracy.

Fig. 5.43(a) shows the effect of ϵ on the JUNO framework, as a parameter to maintain a trade-off between exploration and exploitation, where the random action a_t is chosen with the probability of ϵ. Note that $\epsilon = [1, 0.1]$ means that ϵ is initialized with 1, gradually decreasing with time by $\Delta\epsilon = \dfrac{\epsilon_{max} - \epsilon_{min}}{N_{epoch}}$ to 0.1, with $N_{epoch} = 100$, which is the best strategy according to the results of Fig. 5.43(a). Moreover, Fig. 5.43(b) illustrates the maximum step h_{max} that the guide-policy is initially used, where $h_{max} = 0$ represents the conventional

RL model with no guide-policy. As shown in Fig. 5.43(b), larger h_{max} results in the lower location error.

To illustrate the effectiveness of the proposed framework, we compare it with four baseline models: (i) Weighted Least Square (WLS) anchor selection [101]; (ii) Geometric Dilution of Precision (GDOP) anchor selection [103]; (iii) Nearest Neighbor Node Selection (NN-NS), where the mobile user is localized by N_r number of nearest beacons, and; (iv) Random Node Selection (RNS), where a set of UWB beacons are randomly selected for localization. Fig. 5.43(c) compares the Empirical Cumulative Distribution Function (ECDF) of different frameworks. According to the results shown in Fig. 5.43(c), the location error caused by the proposed JUNO framework is considerably lower than that of its counterparts. Finally, in Fig. 5.44 we consider a random trajectory in a (24×15) m^2 rectangular indoor environment, and compare the ground truth with the estimated trajectories by our proposed JUNO framework and two other baselines. As shown in Fig. 5.44, the estimated path by our proposed framework in the most points closely follows that of the ground truth.

Finally, we use the Root Mean Squared Error (RMSE), which is a generally used performance metric within the localization domain, calculated as follows

$$RMSE = \sqrt{\frac{\sum\limits_{t=1}^{N}(L_t - L_{est,t})^2}{N}}, \tag{211}$$

where $L_t = (x_t, y_t)$ and $L_{est,t} = (x_{es,t}, y_{es,t})$ represent the exact and the estimated location of the user at time t, and N denotes the total number of steps once the proposed JUNO framework reaches the steady state. The proposed JUNO framework achieves the localization error of 0.32 m, while the CNN [207] and LSTM-based [208] localization frameworks track a mobile user with 0.58, 0.48 location errors, respectively.

5.7 Conclusion

In this Chapter, we targeted the problem of connection scheduling in indoor environment consisting of FAPs, operating in the open access mode, and mobile users with the ability of establishing D2D communication. We showed that the proposed MAFH scheme in Section 5.1 reduces the energy consumption of users by limiting high speed clients to download desired contents through FAPs instead of serving through the D2D communication. Moreover, the connecting time in low speed clients is expanded by establishing D2D connection between only users who are approaching together. Concerning high speed clients, the proposed scheme selects the most appropriate target FAP among all candidates by considering the RSSI to extend the time interval between two consecutive handovers. Given an optimal indoor connection scheduling framework, we focused on localization techniques to track users in indoor venues. By focusing on BLE technology, we introduced two analytical solutions in Section 5.2 and 5.3 to increase the location accuracy using the phase difference of the received signal. In Section 5.4, a CNN-based AoA estimation was proposed where the MUSIC algorithm was deployed to extract angle of the BLE signal. Then, our focus was shifted to UWB technology, where in Section 5.5, we developed a DRL-based anchor selection framework to provide high location accuracy, while maintaining a balance between the remained battery of UWB beacons. Finally, Section 5.6 introduced an JSRL model to accelerate the learning process.

Chapter 6

Conclusion and Future Direction

To establish an efficient communication system with high availability and minimal latency, there is a need to overcome limitations in MEC networks. Despite considerable advancements in MEC, there are still challenges posed by the rapid and remarkable increase in global mobile data traffic. This growth is primarily driven by intelligent IoT devices and edge devices, presenting a significant obstacle for wireless networks in the foreseeable future. MEC has emerged as a promising solution to mitigate this challenge by facilitating low-latency communication and reducing network traffic over the backhaul. By storing popular multimedia content in close proximity to end-users, MEC aims to improve the QoS. The main focus throughout this Ph.D. thesis was on integrated and heterogenous MEC networks. We first presented an review of different caching strategies, popularity prediction models, and indoor localization techniques. Then we focused on three important research objectives associated with MEC networks; i.e., connection scheduling, coded/unncoded content placement schemes, and developing mobility-aware indoor/outdoor MEC networks. In the following, we first summarize the thesis contributions, and then discuss potential directions for future research.

6.1 Summary of Contributions

The thesis contributions can be summarized as follows:

- **Autonomous Connection Scheduling and Uncoded Content Placement:** In chapter 3, we targeted the problem of autonomous connection scheduling using RL-based architectures by introducing the CQN-CS framework. Given an optimal connection scheduling framework, we focused on data-driven popularity prediction models adapted with uncoded content placement in MEC networks. We first proposed the attention-based TEDGE caching framework, which is based on ViT network. Despite the fact that the TEDGE framework achieves a high level of classification accuracy, it was intricate and demands a significant number of parameters in order to achieve such accuracy. Then, we presented a low-complex and parallel ViT-CAT fusion architecture to predict the Top-K popular contents in MEC networks. One significant limitation of supervised popularity prediction models is the requirement for manual labeling of contents as popular or unpopular by investigating users' past behavior, which can be a time-intensive task. Therefore, we proposed two self-supervised learning algorithms, i.e., CLSA and CoPo frameworks, to predict the dynamic content popularity in a MEC network in a self-supervised manner.
- **Popularity Prediction and Coded/Uncoded Content Placement:** In chapter 4, we introduced the CCUF framework, which aims to optimize an integrated and dynamic cellular network by maximizing the number of requests served through caching nodes. To enhance cache diversity and enable the storage of different content segments in neighboring FAPs, we utilized a two-phase clustering technique for FAP formation and UAVs deployment. We formulated the success probability of the CCUF framework and employed a cluster-centric approach where multimedia contents were encoded based on their popularity profiles. We also determined the optimal number of fully stored contents to leverage CoMP technology and improve inter-cell interference. Through simulations and by considering the best value of α, the proposed CCUF framework demonstrated increased cache-hit ratio, SINR, and cache diversity, while reducing users' access delay and cache redundancy.

  Additionally, we presented the MTEC framework, which is an efficient multi-content time-series popularity prediction model designed for cluster-centric MEC networks. Existing data-driven caching strategies were not effective for coded/uncoded content placement due to the lack of predicted request probabilities. To address this challenge, we developed a Transformer-based architecture with multiple models. This architecture not only predicts the upcoming Top-K popular content but also estimates the request patterns for multiple contents simultaneously. This information is then used to determine which contents should be stored partially or completely. Simulation results demonstrated that the proposed MTEC caching scheme outperformed state-of-the-art approaches in terms of cache-hit ratio and transferred byte volume.

- **Indoor D2D-enabled MEC Network:** In chapter 5, our objective was to address the issue of connection scheduling in indoor environments that consist of FAPs operating in open access mode, as well as mobile users capable of establishing D2D communication. We presented a solution called MAFH, which aims to reduce energy consumption for users. This is achieved by restricting high-speed clients from utilizing D2D communication and instead encouraging them to download content through FAPs. For low-speed clients, we extend their connection time by establishing D2D connections only between users who are in close proximity. In the case of high-speed clients, our scheme selects the most suitable FAP based on RSSI to increase the interval between consecutive handovers.

  After addressing the optimal indoor connection scheduling framework, we shifted our focus to localization techniques for tracking users within indoor venues. Specifically, we explored the use of BLE technology and introduced two analytical solutions that leverage the phase difference of received signals to improve location accuracy. Additionally, we proposed a CNN-based AoA estimation method, which utilized the MUSIC algorithm to extract the angle of the BLE signal. Subsequently, we turned our attention to UWB technology and developed a DRL-based framework for selecting anchors. This framework aimed to achieve high location accuracy while considering the remaining battery levels of UWB beacons. Finally, we introduced a JSRL model, which helped expedite the learning process.

6.2 Future Direction

Our future directions are as follows:

- A fruitful direction for future research is to find the optimum and time-varying height for each UAV as a function of the remaining battery life of UAVs and the number of users that are served in their transmission coverage.
- It is worth mentioning that wind disturbances can have destructive impacts on the flying path, flying speed, and/or position of UAVs in the low attitude environment [217–219]. The wind effect on UAVs can be viewed as an energy transfer process, changing the flight states of UAVs [223]. Given the complexities involved in the design of cache-enabled UAV-aided cellular networks, however, the effects of wind on flight states of UAVs have not yet been considered within this context [220–222]. This could be another direction for future research to investigate the effect of wind on the placement and path trajectory of cache-enabled UAV-aided cellular networks.
- In DQLEL framework, as the first step towards development of a fully autonomous agent for UWB-based indoor localization, we considered one active user in each time slot. With the emphasis on the multiple access technologies, our future research involves the deployment of a DRL-based localization framework capable of localizing multiple users in each time slot.
- Another future direction is to investigate the security factor and the effect of mobility patterns on the performance of the indoor D2D-enabled MEC networks.
- Furthermore, as the primary focus of the CLSA framework was to introduce an innovative framework for predicting popularity and enhancing content placement, significant emphasis has been placed on this aspect. However, our fruitful direction for future research lies in integrating the proposed CLSA popularity prediction framework with our recent research [4] to explore the collaboration among nearby edge devices for storing distinct content, with a problem formulation that encompasses various performance

factors, including UAVs' energy consumption and user latency within the network. This integration holds promise for advancing the field and addressing broader challenges related to content caching and network optimization.

- Graph Neural Networks (GNNs) have shown promise in the context of MEC networks for content prediction. GNNs are a type of neural network architecture specifically designed to operate on graph-structured data, making them suitable for modeling and predicting relationships between various entities in a network. In the context of MEC networks, GNNs can be used to capture the complex relationships between different content items, users, and caching nodes in the network. By representing the network as a graph, where nodes represent content items, users, and caching nodes, and edges represent relationships or interactions between them, GNNs can learn to extract meaningful features and patterns from this graph structure.

Bibliography

[1] Z. Hajiakhondi-Meybodi, A. Mohammadi and J. Abouei, “Deep Reinforcement Learning for Trustworthy and Time-Varying Connection Scheduling in a Coupled UAV-Based Femtocaching Architecture,” *IEEE Access*, vol. 9, pp. 32263-32281, Feb. 2021.

[2] Z. HajiAkhondi-Meybodi, J. Abouei, M. Jaseemuddin and A. Mohammadi, “Mobility-Aware Femtocaching Algorithm in D2D Networks Based on Handover,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10188-10201, June 2020.

[3] Z. Hajiakhondi-Meybodi, M. Hou and A. Mohammadi, “Content Placement in a Cluster-Centric Mobile Edge Caching Network," *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Shanghai, China, 2022, pp. 1-5.

[4] Z. HajiAkhondi-Meybodi, A. Mohammadi, J. Abouei, and M. Hou. “Joint Transmission Scheme and Coded Content Placement in Cluster-centric UAV-aided Cellular Networks,” *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11098-11114, July 2022.

[5] Z. Hajiakhondi Meybodi, A. Mohammadi, E. Rahimian, S. Heidarian, J. Abouei, and K. N. Plataniotis, “TEDGE-Caching: Transformer-based Edge Caching Towards 6G Networks," *IEEE International Conference on Communication,* Seoul, 2022, pp. 613-618.

[6] Z. HajiAkhondi-Meybodi, A. Mohammadi, M. Hou, J. Abouei, and K. N. Plataniotis, “ViT-Cat: Parallel Vision Transformers With Cross Attention Fusion for Popularity Prediction in MEC Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5.

[7] Z. HajiAkhondi-Meybodi, A. Mohammadi, J. Abouei, and K. N. Plataniotis, “CoPo: Self-supervised Contrastive Learning for Popularity Prediction in MEC Networks,” *IEEE Digital Signal Processing*, June 2023.

[8] Z. Hajiakhondi-Meybodi, A. Mohammadi, J. Abouei, and K. N. Plataniotis, “CLSA: Contrastive Learning-based Survival Analysis for Popularity Prediction in MEC Networks,” *Accepted in IEEE Internet of Things Journal*, March 2023.

[9] Z. Hajiakhondi-Meybodi, A. Mohammadi, M. Hou, E. Rahimian, S. Heidarian, J. Abouei, and K. N. Plataniotis, “Multi-Content Time-Series Popularity Prediction with Multiple-Model Transformers in MEC Networks,” *arXiv preprint arXiv:2210.05874*, Oct. 2022.

[10] Z. HajiAkhondi-Meybodi, M. S. Beni, K. N. Plataniotis, and A. Mohammadi “Bluetooth Low Energy-based Angle of Arrival Estimation via Switch Antenna Array for Indoor Localization,” *International Conference on Information Fusion*, July 2020.

[11] Z. HajiAkhondi-Meybodi, M. S. Beni, A. Mohammadi, and K. N. Plataniotis, “Bluetooth Low Energy-based Angle of Arrival Estimation in Presence of Rayleigh Fading,” *IEEE International Conference on Systems, Man, and Sybernetics (SMC)*, Oct. 2020.

[12] Z. HajiAkhondi-Meybodi, M. Salimibeni, A. Mohammadi, and K. N. Plataniotis, “Bluetooth Low Energy and CNN-Based Angle of Arrival Localization in Presence of Rayleigh Fading,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2021, pp. 7913-7917.

[13] Z. HajiAkhondi-Meybodi, A. Mohammadi, M. Hou, and K. N. Plataniotis, “DQLEL: Deep Q-Learning for Energy-OptimizedLoS/NLoS UWB Node Selection,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 2532-2547, 2022.

[14] Z. Hajiakhondi-Meybodi, M. Hou and A. Mohammadi, “JUNO: Jump-Start Reinforcement Learning-based Node Selection for UWB Indoor Localization," *IEEE Global Communications Conference*, Rio de Janeiro, Brazil, 2022, pp. 6194-6199.

[15] M. Salimibeni, and Z. Hajiakhondi Meybodi, P. Malekzadeh, M. Atashi, K. N. Plataniotis, A. Mohammadi “IoT-TD: IoT Dataset for multiple model BLE-based indoor localization/tracking,” *European Signal Processing Conference (EUSIPCO)*, Amsterdam, Netherlands, 2021, pp. 1697-1701.

[16] M. Atashi, P. Malekzadeh, M. Salimibeni, Z. Hajiakhondi Meybodi, K. N. Plataniotis, A. Mohammadi “Orientation-matched multipe modeling for RSSI-based indoor localization via BLE sensors,” *European Signal Processing Conference (EUSIPCO)*, Amsterdam, Netherlands, 2021, pp. 1697-1701.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, “FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[18] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, “A Survey of Device-to-Device Communications: Research Issues and Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2133-2168, thirdquarter 2018.

[19] F. Cheng, G. Gui, N. Zhao, Y. Chen, J. Tang, and H. Sari, “UAV-Relaying-Assisted Secure Transmission With Caching,” *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3140-3153, May 2019.

[20] N. Zhao, F. Cheng, F. R. Yu, J. Tang, Y. Chen, G. Gui, and H. Sari, “Caching UAV Assisted Secure Transmission in Hyper-Dense Networks Based on Interference Alignment,” *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 2281-2294, May 2018.

[21] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K. K. R. Choo, “Neural-Blockchain-Based Ultrareliable Caching for Edge-Enabled UAV Networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5723-5736, Oct. 2019.

[22] H. Wu, X. Tao, N. Zhang, and X. Shen, “Cooperative UAV Cluster-Assisted Terrestrial Cellular Networks for Ubiquitous Coverage,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2045-2058, Sept. 2018.

[23] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, “Energy-Aware Dynamic Resource Allocation in UAV Assisted Mobile Edge Computing Over Social Internet of Vehicles,” *IEEE Access*, vol. 6, pp. 56700-56715, 2018.

[24] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, “Caching in the Sky: Proactive Deployment of Cache-Enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046-1061, May 2017.

[25] M. Rajabizadeh and J. Abouei, “An Efficient Femtocell-to-Femtocell Handover Decision Algorithm in LTE Femtocell Networks,” *in Proc. IEEE Iranian Conference on Electrical Engineering*, May 2015, pp. 213-218.

[26] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K. K. R. Choo, “Neural-Blockchain-Based Ultrareliable Caching for Edge-Enabled UAV Networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5723-5736, Oct. 2019.

[27] Y. Zeng, J. Xu and R. Zhang, “Energy Minimization for Wireless Communication With Rotary-Wing UAV,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329-2345, April 2019.

[28] S. Ahmed, M. Z. Chowdhury and Y. M. Jang, “Energy-Efficient UAV-to-User Scheduling to Maximize Throughput in Wireless Networks,” *IEEE Access*, vol. 8, pp. 21215-21225, 2020.

[29] D. Athukoralage, I. Guvenc, W. Saad and M. Bennis, “Regret Based Learning for UAV Assisted LTE-U/WiFi Public Safety Networks,” *IEEE Global Communications Conference (GLOBECOM)*, Washington, Feb. 2017 pp. 1-7.

[30] S. Zhu, L. Gui, N. Cheng, F. Sun and Q. Zhang, “Joint Design of Access Point Selection and Path Planning for UAV-Assisted Cellular Networks,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 220-233, Jan. 2020.

[31] Z. Askari, J. Abouei, M. Jaseemuddin and A. Anpalagan, “Energy-Efficient and Real-Time NOMA Scheduling in IoMT-Based Three-Tier WBANs,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13975-13990, 15 Sept.15, 2021.

[32] W. Li, C. Wang, D. Li, B. Hu, X. Wang, and J. Ren, “Edge Caching for D2D Enabled Hierarchical Wireless Networks with Deep Reinforcement Learning,” *Wireless Communications and Mobile Computing*, vol. 2019, Feb. 2019.

[33] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-Agent Reinforcement Learning for Efficient Content Caching in Mobile D2D Networks," *IEEE Trans. on Wireless Communications*, vol. 18, no. 3., pp. 1610-1622, Mar. 2019.

[34] C. Jayasundara, A. Nirmalathas, E. Wong, and N. Nadarajah, "Popularity-Aware Caching Algorithm for Video-on-Demand Delivery over Broadband Access Networks," *Proc. IEEE Global Telecommunications Conference*, Dec. 2010, pp. 1-5.

[35] Y. Wu, D. Wu, L. Yang, and S. Xu, "Incentive-based Cluster Formation for D2D Multicast Content Sharing," *Proc. Asia-Pacific Conference on Communications*, Nov. 2018, pp. 125-130.

[36] L. Fang, R. Zhang, X. Cheng, J. Xiao and L. Yang, "Cooperative content download-and-share: Motivating D2D in cellular networks," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1831-1834, Aug. 2017.

[37] S. Noreen and N. Saxena and A. Roy, "Discount interference pricing mechanism for data offloading in D2D communications," *IEEE Communications Letters*, vol. 22, no. 8, pp. 1688-1691, Aug. 2018.

[38] M. Ji and G. Caire and A. Molisch, "Fundamental Limits of Caching in Wireless D2D Networks," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849-869, Feb. 2016.

[39] K. Poularakis, and L. Tassiulas, "Code, Cache and Deliver on the Move: A Novel Caching Paradigm in Hyper-Dense Small-Cell Networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 675-687, 1 Mar. 2017.

[40] N. Golrezaei, A. F. Molisch, A. G. Dimakis and G. Caire, "Femtocaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142-149, Apr. 2013.

[41] T. Wang and L. Song and Z. Han, "Dynamic Femtocaching for Mobile Users," *Proc. IEEE Wireless Communications and Networking Conference*, Mar. 2015, pp. 861-865.

[42] E. Recayte, F. Lazaro, and G. Liva, "Caching at the Edge with Fountain Codes," in Proc. *Advanced Satellite Multimedia Systems Conference and the Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Berlin, Oct. 2018, pp. 1-6.

[43] D. Ko, B. Hong, and W. Choi, "Probabilistic Caching Based on Maximum Distance Separable Code in a User-Centric Clustered Cache-Aided Wireless Network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1792-1804, Mar. 2019.

[44] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[45] Z. Chen, J. Lee, T. Q. S. Quek and M. Kountouris, “Cooperative Caching and Transmission Design in Cluster-Centric Small Cell Networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3401-3415, May 2017.

[46] A. Liu, and V. K. N. Lau, “Cache-Enabled Opportunistic Cooperative MIMO for Video Streaming in Wireless Systems,” *IEEE Transactions Signal Processing*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[47] Y. Yu, T. Hsieh, and A. Pang, “Millimeter-Wave Backhaul Traffic Minimization for CoMP Over 5G Cellular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4003–4015, Apr. 2019.

[48] P. Lin, Q. Song, and A. Jamalipour, “Multidimensional Cooperative Caching in CoMP-Integrated Ultra-dense Cellular Networks,” *IEEE Transactions Wireless Communication*, vol. 19, no. 3, pp. 1977–1989, Mar. 2020.

[49] H. Li, C. Yang, X. Huang, N. Ansari, and Z. Wang, “Cooperative RAN Caching based on Local Altruistic Game for Single and Joint Transmissions,” *IEEE Communication Letter*, vol. 21, no. 4, pp. 853–856, Apr. 2017.

[50] P. Lin, K. S. Khan, Q. Song, and A. Jamalipour, “Caching in Heterogeneous Ultradense 5G Networks: A Comprehensive Cooperation Approach,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 22–32, Jun. 2019.

[51] A. Giovanidis, and A. Avranas, “Spatial multi-LRU caching for wireless networks with coverage overlaps,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, no. 1, pp. 403-405, June 2016.

[52] K. N. Doan, T. Van Nguyen, T. Q. S. Quek, and H. Shin, “Content-Aware Proactive Caching for Backhaul Offloading in Cellular Network,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3128-3140, May 2018.

[53] L. Ale, N. Zhang, H. Wu, D. Chen and T. Han, “Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Network,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520-5530, June 2019.

[54] Q. Fan, X. Li, J. Li, Q. He, K. Wang and J. Wen, “PA-Cache: Evolving Learning-Based Popularity- Aware Content Caching in Edge Networks,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1746-1757, June 2021.

[55] C. Zhang et al., “Toward Edge-Assisted Video Content Intelligent Caching With Long Short-Term Memory Learning,” *IEEE Access*, vol. 7, pp. 152832-152846, 2019.

[56] S. Rathore, J. H. Ryu, P. K. Sharma and J. H. Park, “DeepCachNet: A Proactive Caching Framework Based on Deep Learning in Cellular Networks,” *IEEE Network*, vol. 33, no. 3, pp. 130-138, May/June 2019.

[57] Y. Lin, C. Yen and J. Wang, “Video Popularity Prediction: An Autoencoder Approach With Clustering,” *IEEE Access*, vol. 8, pp. 129285-129299, 2020.

[58] C. Zhong, M. C. Gursoy and S. Velipasalar, “Deep Reinforcement Learning-Based Edge Caching in Wireless Networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 48-61, March 2020.

[59] P. Wu, J. Li, L. Shi, M. Ding, K. Cai and F. Yang, "Dynamic Content Update for Wireless Edge Caching via Deep Reinforcement Learning," *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1773-1777, Oct. 2019.

[60] Y. Wang, Y. Li, T. Lan and V. Aggarwal, "DeepChunk: Deep Q-Learning for Chunk-Based Caching in Wireless Data Processing Networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1034-1045, Dec. 2019.

[61] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao and M. S. Hossain, "Mobility-Aware Proactive Edge Caching for Connected Vehicles Using Federated Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341-5351, Aug. 2021.

[62] K. C. Tsai, L. Wang and Z. Han, "Mobile Social Media Networks Caching with Convolutional Neural Network," *IEEE Wireless Communications and Networking Conference Workshops*, 2018, pp. 83-88.

[63] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim and C. S. Hong, "Deep Learning Based Caching for Self-Driving Cars in Multi-Access Edge Computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2862-2877, May 2021.

[64] H. Mou, Y. Liu and L. Wang, "LSTM for Mobility Based Content Popularity Prediction in Wireless Caching Networks," *IEEE Globecom Workshops*, 2019, pp. 1-6.

[65] A. Vaswani, N.Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, pp. 5998-6008. 2017.

[66] M. Ozcan, A. Fuad, and G. Haluk, "Accurate and Precise Distance Estimation for Noisy IR Sensor Readings Contaminated by Outliers," *Elsevier: Measurement*, vol. 156, pp. 1–13, Feb. 2020.

[67] R. Carotenuto, M. Merenda, D. Iero, and F. G. D. Corte, "Mobile Synchronization Recovery for Ultrasonic Indoor Positioning," *Sensors*, vol. 20, no. 3, pp. 702–727, Jan. 2020.

[68] M. Maheepala, A. Z. Kouzani, and M. A. Joordens, "Light-based Indoor Positioning Systems: A Review," *IEEE Sensors Journal*, Jan. 2020.

[69] G. W. Samu, and P. Kadam, "Survey on Indoor Localization: Evaluation Performance of Bluetooth Low energy and Fingerprinting based Indoor Localization System," *International Journal of Computer Engineering and Technology (IJCET)*, vol. 8, no. 6, pp. 23–35, Dec. 2017.

[70] S. Weiguang, J. Du, X. Cao, Y. Yu, Y. Cao, S. Yan, and C. Ni, "IKULDAS: An Improved kNN-Based UHF RFID Indoor Localization Algorithm for Directional Radiation Scenario," *Sensors*, vol. 19, no. 4, pp. 968–986, Jan. 2019.

[71] A. Loganathan, N. S. Ahmad, and P. Goh, "Self-Adaptive Filtering Approach for Improved Indoor Localization of a Mobile Node with Zigbee-Based RSSI and Odometry," *Sensors*, vol. 19, no. 21, pp. 4748–4773, Jan. 2019.

[72] G. Huang, Z. Hu, J. Wu, H. Xiao, and F. Zhang, "WiFi and Vision Integrated Fingerprint for Smartphone-Based Self-Localization in Public Indoor Scenes," *IEEE Internet of Things Journal*, Feb. 2020.

[73] P. Malekzadeh, A. Mohammadi, M. Barbulescu, and K. N. Plataniotis, “STUPEFY: Set-Valued Box Particle Filtering for Bluetooth Low Energy-Based Indoor Localization,” *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1773–1777, Dec. 2019.

[74] J. J. P. Solano, S. Ezpeleta, and J. M. Claver, “Indoor localization using time difference of arrival with UWB signals and unsynchronized devices,” *Elsevier: Ad Hoc Networks*, vol. 99, pp. 1–11, Mar. 2020.

[75] M. Atashi, M. Salimibeni, P. Malekzadeh, M. Barbulescu, K. N. Plataniotis, and A. Mohammadi, “Multiple model BLE-based tracking via validation of RSSI fluctuations under different conditions,” *International Conference on Information Fusion (FUSION)*, July 2019, pp. 1–6.

[76] S. Mehryar, P. Malekzadeh, S. Mazuelas, P. Spachos, K. N. Plataniotis, and A. Mohammadi, “Belief condensation filtering for RSSI-based state estimation in indoor localization,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8385–8389.

[77] S. Monfared, T. Nguyen, L. Petrillo, P. De Doncker, and F. Horlin, “Experimental Demonstration of BLE Transmitter Positioning Based on AOA Estimation,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Bologna, Dec. 2018, pp. 856-859.

[78] P. Spachos, I. Papapanagiotou and K. N. Plataniotis, “Microlocation for Smart Buildings in the Era of the Internet of Things: A Survey of Technologies, Techniques, and Approaches," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 140–152, Sep. 2018.

[79] L. Cheng, Y. Wu, J. Zhang and L. Liu, “Subspace Identification for DOA Estimation in Massive/Full-Dimension MIMO Systems: Bad Data Mitigation and Automatic Source Enumeration," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 5897-5909, Nov. 2015.

[80] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, Ch. Yuen, R. Raulefs, and E. Aboutanios, “Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327-1346, Secondquarter 2017.

[81] J. Shen, B. Huang, X. Kang, B. Jia and W. Li, “Localization of Access Points based on the Rayleigh Lognormal Model," *IEEE Wireless Communications and Networking Conference*, 2018, pp. 1-6.

[82] X. Wang, L. Gao and S. Mao, “BiLoc: Bi-modal Deep Learning for Indoor Localization with Commodity 5GHz WiFi," *IEEE access*, vol. 5, pp. 4209-4220, 2017.

[83] J. He, Y. Geng, F. Liu and C. Xu, “CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment," *EEE Sensors Journal*, vol. 14, no. 11, pp. 3766-3774, Nov. 2014.

[84] C. Zhang, X. Bao, Q. Wei, Q. Ma, Y. Yang, and Q. Wang, “ Kalman filter for UWB positioning in LOS/NLOS scenarios," *in Proc. International Conference*

on Ubiquitous Positioning, Indoor Navigation and Location Based Services, Nov. 2016, pp. 73-78.

[85] R. Exel and T. Bigler, “oA ranging using subsample peak estimation and equalizer-based multipath reduction," *in Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Istanbul, Apr. 2014, pp. 2964-2969.

[86] M. Z. Comiter, M. B. Crouse and H. T. Kung, “A Data-Driven Approach to Localization for High Frequency Wireless Mobile Networks," *in Proc. IEEE GLOBECOM Conference*, Singapore, 2017, pp. 1-7.

[87] M. Comiter, and H. T. Kung, “Localization Convolutional Neural Networks Using Angle of Arrival Images," *in Proc. IEEE Global Communications Conference*, Abu Dhabi, Dec. 2018, pp. 1-7.

[88] A. Khan, S. Wang and Z. Zhu, “Angle-of-Arrival Estimation Using an Adaptive Machine Learning Framework," *IEEE Communications Letters*, vol. 23, no. 2, pp. 294-297, Feb. 2019.

[89] X. Wang, X. Wang, and S. Mao. “CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," *IEEE International Conference on Communications (ICC)*, pp. 1-6, May 2017.

[90] X. Wang, X. Wang and S. Mao, “Deep Convolutional Neural Networks for Indoor Localization with CSI Images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316-327, Mar. 2020.

[91] Z. HajiAkhondi-Meybodi, M. S. Beni, A. Mohammadi, and K. N. Plataniotis, “Bluetooth Low Energy-based Angle of Arrival Estimation in Presence of Rayleigh Fading,” *IEEE International Conference on Systems, Man, and Sybernetics*, 2020.

[92] H. Zhang, J. Hu, H. Zhang, B. Di, K. Bian, Z. Han, and L. Song, “MetaRadar: Indoor Localization by Reconfigurable Metamaterials,” *IEEE Transactions on Mobile Computing*, Dec. 2020.

[93] R. Huang, J. Tao, L. Yang, Y. Xue and Q. Wu, “Robust TDoA Indoor Tracking Using Constrained Measurement Filtering and Grid-Based Filtering,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4895-4899.

[94] F. Zafari, A. Gkelias, and K. K. Leung, “A Survey of Indoor Localization Systems and Technologies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, Apr. 2019.

[95] M. Khalaf-Allah, “Particle Filtering for Three-Dimensional TDoA-Based Positioning Using Four Anchor Nodes,” *Sensors*, vol. 20, pp. 4516–4542, Jan. 2020.

[96] R. Huang, L. Yang, J. Tao and Y. Xue, “KLD Minimization-Based Constrained Measurement Filtering For Two-Step TDoA Indoor Tracking,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*", 2021, pp. 4635-4639.

[97] S. Zhao, X. P. Zhang, X. Cui and M. Lu, “Optimal TOA Localization for Moving Sensor in Asymmetric Network,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7928-7932.

[98] L. Zhang, C. Wang, M. Ma and D. Zhang, “WiDIGR: Direction-Independent Gait Recognition System Using Commercial Wi-Fi Devices,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1178–1191, Feb. 2020.

[99] W. Dai, Y. Shen, and M. Z. Win, “Sparsity-Inspired Power Allocation for Network Localization,” *IEEE International Conference on Communications (ICC)*, pp. 2785-2790, June 2013.

[100] C. Wang, Y. Ning, J. Wang, L. Zhang, J. Wan, Q. He, “Optimized Deployment of Anchors based on GDOP Minimization for Ultra-Wideband Positioning,” *Journal of Spatial Science*, pp. 1–18, Nov. 2020.

[101] A. Albaidhani, A. Morell, and J. L. Vicario, “Anchor Selection for UWB Indoor Positioning,” *Transactions on Emerging Telecommunications Technologies* vol. 30, no. 6, June 2019.

[102] A. Courtay, M. L. Gentil, O. Berder, P. Scalart, S. Fontaine and A. Carer, “Anchor Selection Algorithm for Mobile Indoor Positioning using WSN with UWB Radio,” *IEEE Sensors Applications Symposium (SAS)*, 2019, pp. 1-5.

[103] A. Albaidhani, and A. Alsudani, “Anchor Selection by Geometric Dilution of Precision for an Indoor Positioning System using Ultra-Wide Band Technology,” *IET Wireless Sensor Systems*, 2020.

[104] Y. Zhu, Y. Zhang, F. Yan, L. Shen, and Y. Wu, “Node Selection for Cooperative Localization with NLOS Mitigation in Wireless Sensor Networks,” Proc. in *IEEE Globecom Workshops*, Dec. 2016, pp. 1–6.

[105] H. Zhang, X. Qi, Q. Wei, Q., and L. Liu, “TOA NLOS Mitigation Cooperative Localisation Algorithm based on Topological Unit,” *IET Signal Processing*, vol. 14, no. 10, pp. 765–773, Jan. 2021.

[106] S. Wang, and Y. Zhang, “Convex Hull based Node Selection NLoS Mitigation for Indoor Localization,” Proc. in *IEEE Wireless Communications and Networking Conference*, Apr. 2016, pp. 1–5.

[107] L. Qiu and G. Cao, “Popularity-Aware Caching Increases the Capacity of Wireless Networks,” Accepted in *IEEE Transactions on Mobile Computing*, 2019.

[108] M. Chen, W. Saad, and C. Yin, “Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1504-1517, March 2019.

[109] C. M. Albertsen, “Generalizing the first-difference correlated random walk for marine animal movement data,” *Scientific reports*, vol. 9, no. 1, pp. 4017–4031, Mar. 2019.

[110] B. Jiang, J. Yang, H. Xu, H. Song and G. Zheng, “Multimedia Data Throughput Maximization in Internet-of-Things System Based on Optimization of Cache-Enabled UAV,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3525–3532, Apr. 2019.

[111] M. Chen, W. Saad and C. Yin, “Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks,”

IEEE Transactions on Wireless Communications, vol. 18, no. 3, pp. 1504–1517, Mar. 2019.

[112] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K. K. R. Choo, “Neural-Blockchain-Based Ultrareliable Caching for Edge-Enabled UAV Networks,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5723-5736, Oct. 2019.

[113] A. M. Al-Samman, T. A. Rahman, T. Al-Hadhrami, A. Daho, M. N. Hindia, M. H. Azmi, K. Dimyati, and M. Alazab, “Comparative Study of Indoor Propagation Model Below and Above 6 GHz for 5G Wireless Networks,” *Electronics*, vol. 18, no. 1, 2019.

[114] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch and G. Caire, “FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers,” *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[115] K. Xue, L. Li, F. Yang, H. Zhang, X. Li and Z. Han, “Multi-UAV Delay Optimization in Edge Caching Networks: A Mean Field Game Approach,” *in Proc. Wireless and Optical Communications Conference*, May 2019, pp. 1–5.

[116] B. R. Kiran1, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Perez, “Deep reinforcement learning for autonomous driving: A survey,” *arXiv preprint arXiv*, Feb. 2020.

[117] K. Li, T. Zhang and R. Wang, “Deep Reinforcement Learning for Multi objective Optimization," *IEEE Transactions on Cybernetics*, pp. 1-12, Mar. 2020.

[118] K. V. Moffaert, M. M. Drugan and A. Nowe, “Scalarized multi-objective reinforcement learning: Novel design techniques," *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Singapore, 2013, pp. 191-199.

[119] R. Burachik, C. Y. Kaya, and M. M. Rizvi, “A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets," *Journal of Optimization Theory and Applications*, vol. 162, no. 2, pp. 428-446, Aug. 2014.

[120] X. Lu, L. Xiao, C. Dai and H. Dai, “UAV-Aided Cellular Communications with Deep Reinforcement Learning Against Jamming,” *IEEE Wireless Communications*, vol. 27, no. 4, pp. 48-53, Aug. 2020.

[121] M. A. Aref and S. K. Jayaweera, “Robust Deep Reinforcement Learning for Interference Avoidance in Wideband Spectrum",*IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, Cleveland, OH, USA, 2019, pp. 1-5.

[122] C. Wu, S. Shi, S. Gu, L. Zhang, and X. Gu, “Deep Reinforcement Learning-Based Content Placement and Trajectory Design in Urban Cache-Enabled UAV Networks,” *Wireless Communications and Mobile Computing*, Aug. 2020.

[123] T. Zhang, Y. Wang, Y. Liu, W. Xu and A. Nallanathan, “Cache-enabling UAV Communications: Network Deployment and Resource Allocation,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7470-7483, Nov. 2020.

[124] R. Ahmad, E. A. Sundararajan, and A. Khalifeh, “A survey on femtocell handover management in dense heterogeneous 5G networks,” *Telecommunication Systems*, vol. 75, no. 4, pp.481-507, Dec. 2020.

[125] Y. Wang, H. He, and X. Tan, “Robust Reinforcement Learning in POMDPs with Incomplete and Noisy Observations," *arXiv preprint arXiv:1902.05795*, Feb. 2019.

[126] G. Vallero, M. Deruyck, W. Joseph and M. Meo, “Caching at the edge in high energy-efficient wireless access networks,” *IEEE International Conference on Communications (ICC)*, June 2020, pp. 1-7.

[127] C. Joseph, H. Hong, and J. C. Stein, “Forecasting Crashes: Trading Volume, Past Returns, and Conditional Skewness in Stock Prices,” *Journal of Financial Economics*, vol. 61, no. 3, pp. 345-381, 2001.

[128] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[129] Y. Hong, J. J. F. Martinez, and A. C. Fajardo, “Day-Ahead Solar Irradiation Forecasting Utilizing Gramian Angular Field and Convolutional Long Short-Term Memory,” *IEEE Access*, vol. 8, pp. 18741-18753, 2020.

[130] P. Wu, J. Li, L. Shi, M. Ding, K. Cai and F. Yang, “Dynamic Content Update for Wireless Edge Caching via Deep Reinforcement Learning,” *IEEE Commun. Lett.*, vol. 23, no. 10, pp. 1773-1777, Oct. 2019.

[131] Z. Hajiakhondi-Meybodi, J. Abouei, and A. H. F. Raouf, “Cache Replacement Schemes Based on Adaptive Time Window for Video on Demand Services in Femtocell Networks,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1476-1487, July 2019.

[132] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, and D. Cai, “What to Do Next: Modeling User Behaviors by Time-LSTM,” *IJCAI*, vol. 17, pp. 3602-3608, 2017.

[133] A. Thakkar and R. Lohiya, “Attack Classification of Imbalanced Intrusion Data for IoT network Using Ensemble Learning-based Deep Neural Network,” *IEEE Internet of Things Journal*, 2023.

[134] N. Y. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321-357, 2002.

[135] Movielens100k Dataset, *https://grouplens.org/datasets/movielens/100k/*

[136] Movielens1M Dataset, *https://grouplens.org/datasets/movielens/1m/*

[137] Movielens25M Dataset, *https://grouplens.org/datasets/movielens/25m/*

[138] Jester Dataset, *https://eigentaste.berkeley.edu/dataset/*

[139] Book-Crossing Dataset, *http://www2.informatik.uni-freiburg.de/ cziegler/BX/*

[140] Last.FM Dataset, *https://grouplens.org/datasets/hetrec-2011/*

[141] M. C. Cieslak, A. M. Castelfranco, V. Roncalli, P. H. Lenz, and D.K. Hartline, "T-Distributed Stochastic Neighbor Embedding (T-SNE): A Tool for Eco-Physiological Transcriptomic Analysis," *Marine genomics*, vol. 51, p. 100723, 2020.

[142] S. Bhandari, N. Ranjan, P. Khan, H. Kim, Y. S. Hong, "Deep Learning-based Content Caching in the Fog Access Points," *Electronics*, vol. 10, no. 4, pp. 512, 2021.

[143] C. Selvi, and E. Sivasankar. "A novel Adaptive Genetic Neural Network (AGNN) model for recommender systems using modified k-means clustering approach," *Multimedia Tools and Applications*, vol. 78, pp. 14303-14330, June 2019.

[144] D. Bahri, H. Jiang, Y. Tay, D. Metzler, "Scarf: Self-Supervised Contrastive Learning using Random Feature Corruption," *arXiv preprint arXiv:2106.15147*, 2022.

[145] M. Afshang, H. S. Dhillon and P. H. J. Chong"Fundamentals of Cluster-Centric Content Placement in Cache-Enabled Device-to-Device Networks," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2511-2526, June 2016.

[146] S. Kumar, and R. Tiwari, "Optimized content centric networking for future internet: dynamic popularity window based caching scheme," *Computer Networks*, vol. 179, pp. 107434, 2020.

[147] A. H. Fahim Raouf and J. Abouei, "Cache Replacement Scheme Based on Sliding Window and TTL for Video on Demand," *IEEE Iranian Conference on Electrical Engineering*, 2018, pp. 499-504.

[148] G. Einziger and R. Friedman, "TinyLFU: A highly efficient cache admission policy," *IEEE Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2014, pp. 146–153.

[149] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao and X. Shen, "Cooperative Edge Caching in User-Centric Clustered Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1791-1805, Aug. 2018.

[150] K. Xue, L. Li, F. Yang, H. Zhang, X. Li and Z. Han, "Multi-UAV Delay Optimization in Edge Caching Networks: A Mean Field Game Approach," *Wireless and Optical Communications Conference*, Dec. 2019, pp. 1-5.

[151] F. Zhang, G. Han, L. Liu, M. Martínez-García and Y. Peng, "Joint Optimization of Cooperative Edge Caching and Radio Resource Allocation in 5G-Enabled Massive IoT Networks," Accepted in *IEEE Internet of Things Journal*, Mar. 2021.

[152] Y. Choi, C. S. Kim and S. Bahk, "Flexible Design of Frequency Reuse Factor in OFDMA Cellular Networks," *IEEE International Conference on Communications*, Istanbul, 2006, pp. 1784-1788.

[153] Wireless Communication. Available online: http://www.wirelesscommunication.nl/ reference/chaptr04/cellplan/reuse.html (accessed on 21-06-11).

[154] A. M. Al-Samman, T. A. Rahman, T. Al-Hadhrami, A. Daho, M. N. Hindia, M. H. Azmi, K. Dimyati, and M. Alazab, "Comparative Study of Indoor Propagation Model Below and Above 6 GHz for 5G Wireless Networks," *Electronics*, vol. 18, no. 1, Jan. 2019.

[155] A. Al-Hourani, S. Kandeepan and S. Lardner, "Optimal LAP Altitude for Maximum Coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569-572, Dec. 2014.

[156] M. Alzenad, A. El-Keyi and H. Yanikomeroglu, "3-D Placement of an Unmanned Aerial Vehicle Base Station for Maximum Coverage of Users With Different QoS Requirements," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 38-41, Feb. 2018.

[157] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647-1650, Aug. 2016.

[158] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881-892, July 2002.

[159] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574-7589, Nov. 2017.

[160] Y. Zhou, N. Cheng, N. Lu and X. S. Shen, "Multi-UAV-Aided Networks: Aerial-Ground Cooperative Vehicular Networking Architecture," *IEEE Vehicular Technology Magazine*, vol. 10, no. 4, pp. 36-44, Dec. 2015.

[161] C. Wu, S. Shi, S. Gu, L. Zhang, and X. Gu, "Deep Reinforcement Learning-Based Content Placement and Trajectory Design in Urban Cache-Enabled UAV Networks," *Wireless Communications and Mobile Computing*, Aug. 2020.

[162] S. T. Muntaha, S. A. Hassan, H. Jung and M. S. Hossain, "Energy Efficiency and Hover Time Optimization in UAV-Based HetNets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5103-5111, Aug. 2021.

[163] M. Khan, "Quadcopter flight dynamics," *Int. J. Sci. Technol. Res.*, vol. 3, no. 8, pp. 130–135, 2014.

[164] A. Joukhadar, M. AlChehabi, C. Stoger, and A. Muller, "Trajectory tracking control of a quadcopter UAV using nonlinear control," *Mechanisms and Machine Science*, vol. 58. Amsterdam, The Netherlands: Springer, 2019, pp. 271–285.

[165] M. Alzenad, A. El-Keyi, F. Lagum and H. Yanikomeroglu, "3-D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434-437, Aug. 2017.

[166] M. Mozaffari, W. Saad, M. Bennis, Y. H. Nam and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334-2360, thirdquarter 2019.

[167] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Drone Small Cells in the Clouds: Design, Deployment and Performance Analysis," *IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1-6.

[168] C. M. Albertsen, "Generalizing the First-Difference Correlated Random Walk for Marine Animal Movement Data," *Scientific Reports*, vol. 9, no. 1, pp. 4017–4031, Mar. 2019.

[169] M. Chen, W. Saad, and C. Yin, "Liquid State Machine Learning for Resource and Cache Management in LTE-U Unmanned Aerial Vehicle (UAV) Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1504-1517, Mar. 2019.

[170] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia Data Throughput Maximization in Internet-of-Things System Based on Optimization of Cache-Enabled UAV," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3525-3532, Apr. 2019.

[171] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," *annual international conference on Mobile computing and networking*, pp. 617-628, 2014.

[172] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," *annual international conference on mobile computing and networking*, pp. 65-76, 2015.

[173] C. M. Albertsen, "Generalizing the first-difference correlated random walk for marine animal movement data" *Scientific reports*, vol. 9, no. 1, pp. 4017, 2019.

[174] S. Bai, X. Bai, X. Che, "Window-LRFU: a cache replacement policy subsumes the LRU and window-LFU policies" *Concurrency and Computation: Practice and Experience*, vol. 28, no. 9, pp. 2670-2684, 2016.

[175] E. Recayte, F. Lazaro, and G. Liva, "Caching at the Edge with Fountain Codes" *Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Berlin, Germany, 2018, pp. 1-6.

[176] S. R. Valizadeh, and J. Abouei, "An adaptive distributed coverage optimization scheme in LTE enterprise femtocells" *Iranian Conference on Electrical Engineering (ICEE)*, Tehran, Iran, 2014, pp. 1723-1728.

[177] P. Xu, X. Fang, R. He, Z. Xiang, "An efficient handoff algorithm based on received signal strength and wireless transmission loss in hierarchical cell networks" *Telecommunication Systems*, vol. 52, no. 9, pp. 317-325, 2013.

[178] C. V. Anamuro, N. Varsier, J. Schwoerer and X. Lagrange, "Simple modeling of energy consumption for D2D relay mechanism" *IEEE Wireless Communications*

and Networking Conference Workshops (WCNCW), Barcelona, Spain, 2018, pp. 231-236.

[179] W. Zheng, H. Zhang, X. Chu, and X. Wen, “Mobility robustness optimization in self-organizing LTE femtocell networks” *EURASIP Journal on Wireless Communications and Networking*, 2013, pp. 1-10.

[180] W. Jiang, G. Feng, S. Qin, T. S. P. Yum and G. Cao, “Multi-Agent Reinforcement Learning for Efficient Content Caching in Mobile D2D Networks” *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1610-1622, March 2019.

[181] K. Poularakis and L. Tassiulas, “Code, Cache and Deliver on the Move: A Novel Caching Paradigm in Hyper-Dense Small-Cell Networks” *IEEE Trans. on Mobile Computing*, vol. 16, no. 3, pp. 675-687, March 2017.

[182] A. Calhan, M. Cicioglu, “Handover scheme for 5G small cell networks with non-orthogonal multiple access” *Computer Networks*, vol. 183, pp. 107601, 2020.

[183] P. Hu, Q. Bao and Z. Chen, “Target Detection and Localization Using Non-Cooperative Frequency Agile Phased Array Radar Illuminator,” *IEEE Access*, vol. 7, pp. 111277-111286, Aug. 2019.

[184] G. Kaur, N. Kaur and L. Kansal, “Performance of OSTBC5 through three different channels using zero forcing equalizer in MIMO system," IEEE International Conference on System, Computation, Automation and Networking (ICSCA), July 2018, pp. 1-7.

[185] X. Zhang, W. Chen, W. Zheng, Z. Xia and Y. Wang, “Localization of Near-Field Sources: A Reduced-Dimension MUSIC Algorithm," IEEE Communications Letters, vol. 22, no. 7, pp. 1422-1425, July 2018.

[186] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang and X. Shen, “Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1722-1760, thirdquarter 2020.

[187] T. Wang, W. Heinzelman, and A. Seyedi, “Link Energy Minimization in IR-UWB based Wireless Networks,” *IEEE transactions on wireless communications*, vol. 9, no. 9, pp. 2800-2811, Sep. 2010.

[188] S. Subedi, D. H. Kim, B. H. Kim and J. Y. Pyun, “Improved Smartphone-Based Indoor Localization System Using Lightweight Fingerprinting and Inertial Sensors," *IEEE Access*, vol. 9, pp. 53343-53357, 2021.

[189] R. C. Luo, and T. J. Hsiao, “Indoor Localization System Based on Hybrid Wi-Fi/BLE and Hierarchical Topological Fingerprinting Approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10791-10806, Nov. 2019.

[190] L. Mraz, “Accuracy Considerations for UWB Indoor Tracking in an Industrial Environment", *Available online: https: //www.sewio.net/accuracy-considerations-for-uwb-indoor-tracking-in-an-industrial-environment/*, 2019, (accessed on 1 March 2022).

[191] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, “Ultra wideband indoor positioning technologies: Analysis and recent advances", *Sensors*, vol. 16, no. 5, pp. 707, May 2016.

[192] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh, “Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624-635, Apr. 2018.

[193] X. Lu, L. Xiao, C. Dai and H. Dai, “UAV-Aided Cellular Communications with Deep Reinforcement Learning Against Jamming", *IEEE Wireless Communications*, vol. 27, no. 4, pp. 48-53, Aug. 2020.

[194] M. A. Aref, and S. K. Jayaweera, “Robust Deep Reinforcement Learning for Interference Avoidance in Wideband Spectrum,” Proc. in *IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, Jun. 2019, pp. 1–5.

[195] M. Delamare, D. Fabrice, and R. Boutteau, “A New Dataset of People Flow in an Industrial Site with UWB and Motion Capture Systems," *Sensors*, vol. 20, no.16, pp. 4511, 2020.

[196] C. Hsieh, J. Chen and B. Nien, “Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information,” *IEEE Access*, vol. 7, pp. 33256-33267, 2019.

[197] JL Ba, JR Kiros, and G.E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[198] O. Eriksson, and Ma. Larsson. “Increasingly Complex Environments in Deep Reinforcement Learning," *Diva Portal*, 2019.

[199] Y. Rioual, Y. L. Moullec, J. Laurent, M. I. Khan, and J. P. Diguet, “Design and Comparison of Reward Functions in Reinforcement Learning for Energy Management of Sensor Nodes,” *arXiv preprint arXiv:2106.01114*, 2021.

[200] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot and A. Ashkan, “Cache Content-Selection Policies for Streaming Video Services,” *IEEE International Conference on Computer Communications (INFOCOM)*, 2016, pp. 1-9.

[201] C. Hong, F. Yi and Z. Huang, “Deep-CSA: Deep Contrastive Learning for Dynamic Survival Analysis With Competing Risks,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 4248-4257, Aug. 2022.

[202] I. Uchendu, T. Xiao, Y. Lu, B. Zhu, M. Yan, J. Simon, M. Bennice, C. Fu, C. Ma, J. Jiao, and S. Levine, “Jump-Start Reinforcement Learning,” *arXiv preprint arXiv:2204.02372*, Apr. 2022.

[203] M. Mohammadi, et al., “Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 624-635, April 2018.

[204] E. Marchesini et al., “Discrete Deep Reinforcement Learning for Mapless Navigation,” *IEEE Inter. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 10688-10694.

[205] L. Gen, et al. “Is Q-learning minimax optimal? A tight sample complexity analysis,” *arXiv preprint arXiv:2102.06548*, 2021.

[206] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang and N. El-Sheimy, “Deep Reinforcement Learning (DRL): Another Perspective for Unsupervised Wireless Localization,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6279-6287, July 2020.

[207] D. T. A. Nguyen, et al. “Deep learning-based localization for UWB systems,” *Electronics*, val. 9, no. 10, pp. 1712, 2020.

[208] A. Poulose, D. S. Han, “UWB Indoor Localization Using Deep Learning LSTM Networks,” *Appl. Sci.* 2020, pp. 6290.

[209] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, “Multi-Tenant Cross-Slice Resource Orchestration: A Deep Reinforcement Learning Approach,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2377-2392, Oct. 2019.

[210] Online, https://e2e.ti.com/support/wireless-connectivity/bluetooth-group/bluetooth/f/bluetooth-forum/810266/boostxl-aoa-about-rtls-and-rtls-monitor, *Access Date: 18 October 2023*.

[211] Y. Choi, C. S. Kim and S. Bahk, “Flexible Design of Frequency Reuse Factor in OFDMA Cellular Networks,” *IEEE International Conference on Communications*, Istanbul, 2006, pp. 1784-1788.

[212] F. M. Harper, and J. A. Konstan, “The Movielens Datasets: History and Context,” *ACM transactions on interactive intelligent systems*, vol. 5, no. 4, pp. 1-19, 2015.

[213] Wireless Communication. Available online: http://www.wirelesscommunication.nl/ reference/chaptr04/cellplan/reuse.html (accessed on 21-06-11).

[214] S. Li, J. Xu, M. van der Schaar and W. Li, “Popularity-Driven Content Caching,” *IEEE International Conference on Computer Communications (INFOCOM)*, 2016, pp. 1-9.

[215] M. T. Nguyen, D. H. Le, T. Nakajima, M. Yoshimi and N. Thoai, “Attention-Based Neural Network: A Novel Approach for Predicting the Popularity of Online Content,” *IEEE International Conference on High Performance Computing and Communications*, 2019, pp. 329-336.

[216] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang and N. El-Sheimy, "Deep Reinforcement Learning (DRL): Another Perspective for Unsupervised Wireless Localization," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6279-6287, July 2020.

[217] H. M. Chung, S. Maharjan, Y. Zhang, F. Eliassen and K. Strunz, “Placement and Routing Optimization for Automated Inspection With Unmanned Aerial Vehicles: A Study in Offshore Wind Farm,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3032-3043, May 2021.

[218] E. Kuantama, R. Tarca, S. Dzitac, I. Dzitac, T. Vesselenyi, and I. Tarca, “The design and experimental development of air scanning using a sniffer Quadcopter,” *Sensors*, vol. 19, no. 18, pp. 3849, 2019.

[219] A. Kachroo et al., "Emulating UAV Motion by Utilizing Robotic Arm for mmWave Wireless Channel Characterization," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 10, pp. 6691-6701, Oct. 2021.

[220] N. Nouri, F. Fazel, J. Abouei and K. Plataniotis, "Multi-UAV Placement and User Association in Uplink MIMO Ultra-Dense Wireless Networks," *IEEE Transactions on Mobile Computing*, 2021.

[221] N. Nouri, J. Abouei, A. R. Sepasian, M. Jaseemuddin, A. Anpalagan and K. N. Plataniotis, "3D Multi-UAV Placement and Resource Allocation for Energy-Efficient IoT Communication," *IEEE Internet of Things Journal*, 2021.

[222] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah and C. S. Hong, "Caching in the Sky: Proactive Deployment of Cache-Enabled Unmanned Aerial Vehicles for Optimized Quality-of-Experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046-1061, May 2017.

[223] Z. Wang, L. Duan and R. Zhang, "Adaptive Deployment for UAV-Aided Communication Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4531-4543, Sept. 2019.