IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 20 May 2024; revised 9 October 2024 and 27 December 2024; accepted 30 January 2025.
Date of publication 11 February 2025; date of current version 20 February 2025.

The associate editor coordinating the review of this article and approving it for publication was N. Pappas.

Digital Object Identifier 10.1109/TMLCN.2025.3540747

Federated Learning-Based Collaborative Wideband Spectrum Sensing and Scheduling for UAVs in UTM Systems

SRAVAN REDDY CHINTAREDDY[1] (Graduate Student Member, IEEE), KEENAN ROACH[2], KENNY CHEUNG[2], AND MORTEZA HASHEMI[1] (Senior Member, IEEE)

[1]Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045 USA
[2]Universities Space Research Association (USRA), Washington, DC 20024 USA

CORRESPONDING AUTHOR: S. R. CHINTAREDDY (sravan.ch@ku.edu)

This work was supported in part by the National Aeronautics and Space Administration under Award 80NSSC20M0261; and in part by NSF under Grant 1955561, Grant 2212565, and Grant 2323189.

ABSTRACT In this paper, we propose a data-driven framework for collaborative wideband spectrum sensing and scheduling for networked unmanned aerial vehicles (UAVs), which act as secondary users (SUs) to opportunistically utilize detected ''spectrum holes''. Our overall framework consists of three main stages. Firstly, in the *model training* stage, we explore dataset generation in a multi-cell environment and train a machine learning (ML) model using the federated learning (FL) architecture. Unlike the existing studies on FL for wireless that presume datasets are readily available for training, we propose an end-to-end architecture that directly integrates wireless dataset generation, which involves capturing I/Q samples from over-the-air signals in a multi-cell environment, into the FL training process. To this purpose, we propose a multi-label classification problem for wideband spectrum sensing to detect multiple spectrum holes simultaneously based on the I/Q samples collected locally by the UAVs. In the traditional FL that employs federated averaging (FedAvg) as the aggregating method, each UAV is assigned an equal weight during model aggregation. However, due to the differences in wireless channels observed at each UAV in a multi-cell environment, the received signal powers and collected datasets at different UAV locations could be significantly different, which could degrade the FL performance using equal weights. To address this issue, we propose a proportional weighted federated averaging method (pwFedAvg) in which the aggregating weights are proportional to the received signal powers at each UAV, thereby integrating the intrinsic properties of wireless channels into the FL algorithm. Secondly, in the *collaborative spectrum inference* stage, we propose a collaborative spectrum fusion strategy that is compatible with the unmanned aircraft system traffic management (UTM) ecosystem. In particular, we improve the accuracy of spectrum sensing results by combining the multi-label classification results from the individual UAVs by performing spectrum fusion at a central server. Finally, in the *spectrum scheduling* stage, we leverage reinforcement learning (RL) solutions to dynamically allocate the detected spectrum holes to the secondary users. To evaluate the proposed methods, we establish a comprehensive simulation framework that generates a near-realistic synthetic dataset using MATLAB LTE toolbox by incorporating base station (BS) locations in a chosen area of interest, performing ray-tracing, and emulating the primary user's channel usage in terms of I/Q samples. This evaluation methodology provides a flexible framework to generate *large spectrum datasets* that could be used for developing ML/AI-based spectrum management solutions for aerial devices.

INDEX TERMS UAV-based spectrum sensing, collaborative inference, federated learning (FL), reinforcement learning (RL), UAS traffic management (UTM).

I. INTRODUCTION

UNMANNED aerial vehicles (UAVs) have attracted significant interest from communications and networking, robotics, and control societies for exploring novel applications such as on-demand connectivity, search-and-rescue operations, and situational awareness, to name a few [1].

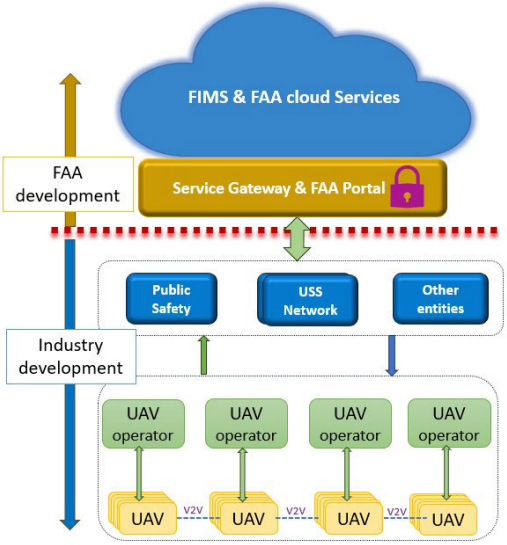


FIGURE 1. Unmanned Aircraft System Traffic Management (UTM) architecture showing the separation between Federal Aviation Administration (FAA) and industry developments; Flight Information Management System (FIMS).

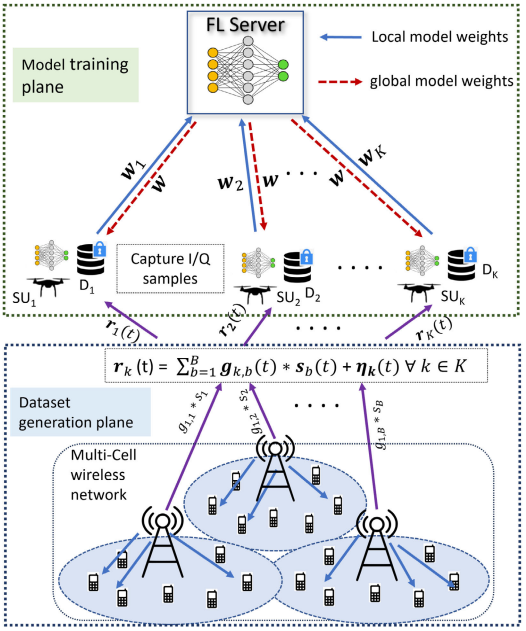


FIGURE 2. Envisioned FL system model in a Multi-cell wireless network with multiple UAVs.

As of April 2024, there were roughly 800,000 registered UAVs in the US alone, positioning UAVs as one of the fastest-growing sectors in the aviation industry [2]. Traditionally, UAVs that are used for recreational purposes are operated under visual line of sight (VLOS) conditions. However, real-world and commercial deployments will most likely be in the form of beyond visual line-of-sight (BVLOS), which provides easier access to remote or hazardous areas, less human intervention, and reduced cost of operation [3]. For safe operations of multiple UAVs under BVLOS conditions, the National Aeronautics and Space Administration (NASA) and Federal Aviation Administration (FAA) are in the process of defining the UTM system [4]. Fig. 1 shows a simplified form of the UTM architecture, highlighting the separation between FAA and industry development and deployment responsibilities for the necessary infrastructure, services, and entities that interact within the UTM ecosystem. In this work, we mainly focus on the hierarchical structure between multiple operators and the UAS service supplier (USS), which assists multiple operators in meeting UTM operational requirements, ensuring safe and efficient utilization of the airspace.

The concept of operations within the UTM architecture [4] highlights the need for spectrum resources to facilitate wireless communications between UAVs, UAV operators, and the USS network. Existing terrestrial mobile networks (for example, 4G LTE and 5G networks) provide significant wireless coverage with relatively low latency, high throughput, and low cost, making the cellular network a good candidate for UAV operation in BVLOS scenarios [5]. However, the proliferation of new wireless services and the demand for higher cellular data rates have significantly exacerbated the spectrum crunch that cellular providers are already experiencing. Therefore, it is essential to develop dynamic spectrum sensing, inference, and sharing solutions for UAV operations in existing licensed and unlicensed spectrum to enable advanced aerial use cases in BVLOS, such as urban air mobility (UAM) and advanced air mobility (AAM) [6], [7].

There exists a multitude of prior works on spectrum management frameworks for ground users [8], [9], [10], [11], [12]. For instance, the authors in [9], [10], and [11] propose deep learning-based wideband spectrum sensing to dynamically detect "spectrum holes". Furthermore, the authors in [12] propose reinforcement learning (RL) techniques for spectrum sharing, assuming that spectrum sensing results are readily available. While these data-driven spectrum management frameworks for ground users are available, they are not directly applicable for UTM-enabled UAV operations, due to several factors, such as the widely different wireless channel models and the overall system architecture [9], [12]. In the context of UAV spectrum sharing systems, the authors in [13] and [14] proposed spatial-spectral sensing (SSS) to develop efficient spectrum sharing policies for UAV communications aimed at improving the overall spectral efficiency (SE). However, the SSS models do not consider the spectrum usage pattern of users under realistic scenarios (e.g., ignoring the I/Q level samples), and/or they consider only a single primary user (PU) or secondary user (SU). Moreover, the problem of joint multi-channel wideband spectrum sensing and scheduling among several SUs has not been fully investigated.

In this paper, we propose a unified and data-driven spectrum sensing and scheduling framework to enable UAVs to effectively share the spectrum with existing primary users. To make our development more concrete and grounded, the problem of joint spectrum sensing and sharing is formulated as energy efficiency (EE) maximization in a wideband

multi-UAV network scenario. Then, we transform the EE optimization problem into a Markov Decision Process (MDP) to maximize the overall throughput of the SUs. At the spectrum sensing stage, we note the inherent hierarchical nature of the UTM architecture with USS (shown in Fig. 1) is a good match for federated learning (FL) based spectrum sensing. For the spectrum scheduling stage, we develop RL-based solutions to enable non-manual and automated spectrum resource allocation.

Particular to the spectrum sensing stage, we propose an FL-based cooperative wideband spectrum sensing across multiple UAVs. To this purpose, we develop a multi-label classification framework to identify spectrum holes based on the observed I/Q samples. Each UAV trains its respective local models using the locally collected datasets and transmits the local model parameters to the central server. Furthermore, we propose a proportional weighted federated averaging (pwFedAvg) method that incorporates the received signal power at each UAV into the FL aggregation algorithm, thereby integrating the dataset generation plane with the FL model training plane, as shown in Fig. 2. Once the training process is completed, all UAVs have an updated global model that predicts spectrum holes. To further enhance the accuracy of the individual spectrum inference results, the predicted spectrum holes from the multi-label classification at each UAV are fused at a central server within the UTM ecosystem. In the spectrum scheduling stage, we develop and implement several RL algorithms, including the standard Q-learning methods to dynamically allocate underutilized spectrum sub-channels to multiple UAVs. We further investigate the performance of the "vanilla" deep Q-Network (DQN) and its variations, including double DQN (DDQN) and DDQN with soft-update.

Furthermore, one of the primary challenges of using machine learning (ML) based methods for spectrum sensing and scheduling approaches is the need for large amounts of training data. The lack of available spectral data in many cases is a significant obstacle, especially for UAV networks that introduce an additional level of complexity for large-scale experimental data collection. To address this gap, we have developed a comprehensive framework for generating spectrum datasets. This framework models LTE waveform generation and propagation channels in any environment of interest, particularly suitable for UTM-enabled UAV applications. Using the generated dataset, we provide a comprehensive set of numerical results to demonstrate the efficacy of the joint FL-based spectrum sensing, spectrum fusion, and RL-based dynamic spectrum allocation to multiple UAVs. In summary, the main contributions of this paper are as follows:

- We develop a spectrum management framework based on the envisioned UTM deployment architecture. To this end, we propose a joint spectrum sensing and scheduling problem for collaborative networked UAVs that operate according to the UTM rules. The joint optimization problem integrates the spectrum sensing results into the spectrum scheduling stage for scenarios with multiple secondary users (i.e., UAVs) and primary users.
- For spectrum sensing, we propose an FL-based solution to enable collaborative model training across distributed UAVs. We propose the pwFedAvg method that integrates the underlying wireless channel conditions into the FL aggregation step. We also provide the convergence analysis results of the proposed pwFedAvg method. Furthermore, we demonstrate the benefits of collaborative spectrum sensing by incorporating a fusion step, which further enhances the performance of wideband spectrum sensing. For the spectrum scheduling stage, we leverage Q-learning-based solutions to allocate the detected spectrum holes to the requesting UAVs.
- We outline a methodology for generating large amounts of I/Q dataset for UAVs in a wide geographical area, considering the effects in a multi-cell multi-path environment by incorporating the base station locations and accurately modeling the environment using ray-tracing methods. Based on the established framework, we provide a comprehensive set of numerical results to analyze the performance of pwFedAvg compared with the traditional FedAvg approach, as well as with centralized and local learning. Our results demonstrate the efficacy of the pwFedAvg method for collaborative spectrum sensing, without the need to transfer all I/Q samples to one location as in central learning.

This paper extends our prior work [15] in which we did not investigate the feasibility of model training using FL methods for UAVs. In contrast, this paper mainly focuses on developing FL-based spectrum sensing by incorporating the wireless datasets captured by multiple UAVs into the FL model training plane, as shown in Fig. 2. Furthermore, we have significantly extended our dataset generation by scaling the size of captured I/Q data samples and increasing the number of reflection and diffraction rays, thereby enhancing the fidelity of emulating the propagation environment. The remainder of this paper is organized as follows. In Section II, we review related works. In Section III, we present the overall system model and problem formulation for FL-based wideband spectrum sensing and collaborative spectrum inference and scheduling. In Section IV, we discuss the dataset generation model and the model training aspects of the FL-based solution to incorporate our proposed pwFedAvg method, followed by a discussion of the convergence analysis of pwFedAvg. In Section V, we present dynamic spectrum allocation using RL. Section VI describes our methodology to generate a synthetic spectrum dataset followed by our numerical results in Section VII. Finally, Section VIII concludes the paper.

II. RELATED WORKS

A. SPECTRUM SENSING AND SHARING FOR UAVs

The authors in [16] address spectrum access and interference management by utilizing SSS for ground-based

device-to-device (D2D) communications [17], [18]. Furthermore, the authors in [14] and [19] extend the usage of SSS to UAVs to opportunistically access the licensed channels that are occupied by the D2D communications of ground users. The UAVs perform SSS to obtain the received signal strength and compare it with a threshold to identify the spectrum occupancy of a particular D2D channel. However, in general, energy-based detection methods would require capturing the entire waveform for a sub-channel to compute the energy and compare it with a predefined threshold. When there are multiple sub-channels, such detection methods repeated for each sub-channel add further time and hardware complexity. Therefore, SSS methods are not directly applicable to wideband spectrum sensing by UAVs to detect multiple spectrum holes *simultaneously*.

In addition to the SSS methods, data-driven deep learning (DL) methods for spectrum sensing have been considered in prior works [20], [21]. To develop multi-channel spectrum sensing using DL, the authors in [9] developed a fast wideband spectrum sensing based on DL. The DL model is based on a convolutional neural network (CNN) that accepts raw I/Q signals and predicts the spectrum holes. The above works consider a single PU, a single SU only, and the channel between the PU and SU is modeled as a Rayleigh fading channel.

Furthermore, there exists extensive research on spectrum-sharing solutions. For example, the authors in [11], [22], [23], and [24] propose the use of RL for dynamic spectrum access in multi-channel wireless networks. Furthermore, the authors in [12], [25], and [26] propose the use of DQN, in which, during each time slot, a single SU decides whether to stay idle or transmit using one of the sub-channels in a multi-channel environment without performing spectrum sensing. While these studies have provided significant insights, they consider one SU only and are well-studied for ground-based communications.

In this paper, we consider a data-driven approach to predict multiple spectrum holes *simultaneously* from the raw I/Q signals captured in a multi-cell multi-path fading environment consisting of multiple PUs and SUs. We incorporate ray-tracing methods to effectively model the dynamic UAV environment instead of assuming a statistical channel model. Furthermore, we employ RL for dynamically allocating resources for the UAVs based on predicted spectrum holes.

B. FL-BASED SPECTRUM SENSING

FL for spectrum sensing has lately gained popularity [27], [28], [29], [30], [31]. The authors in [27], [29] discuss the application of FL for spectrum sensing in cognitive radio environments, where an SU detects the spectrum holes in the PU's spectrum band and utilizes them opportunistically. However, the studies only considered a single PU with multiple SUs within the coverage of the PU. There exists a separate class of research that concentrates on interference management in multi-cell wireless networks and incorporates over-the-air computation in FL [32], [33], [34], [35]. For instance, the authors in [32] study the adverse effects of inter-cell interference on the uplink and downlink local model aggregates and global model updates and propose solutions to mitigate the interference. In contrast to the above-mentioned works, in this paper, we cater to multiple PUs and SUs, incorporate wireless channel characteristics of a multi-cell network at the dataset generation level, and also consider wideband spectrum sensing.

Furthermore, there are several research works on FL for wireless systems that investigate how the convergence of the learning process is affected by the noisy transmissions between the clients and the server [36], [37], [38]. The authors in [39], [40], [41] proposed communication-efficient federated learning that quantizes the model weights before transmission. In particular, the authors in [39] propose a dynamic aggregation for heterogeneous quantization, where different weights are assigned to clients based on their quantization error. These studies often assume that the datasets are readily available to clients and focus primarily on the specific discussions of the model training plane, as illustrated in Fig. 2. Furthermore, these studies consider standard ML datasets, such as CIFAR-10, MNIST, and Shakespeare [37].

Yet, wireless datasets collected by multiple UAVs in a multi-cell environment are significantly complex and different compared to those standard datasets. For instance, data collected at one UAV location may encounter distinct wireless channels, varying numbers of propagation paths, and significantly different received signal power levels compared to the data collected at other locations. This variability underscores the need for tailored approaches to model training within FL frameworks, particularly when dealing with datasets from real-world wireless environments.

In this paper, we demonstrate and evaluate the performance of FL models for wideband spectrum sensing using over-the-air I/Q datasets, as illustrated in Fig. 2. This framework incorporates capturing wireless datasets by UAVs operating in a multi-cell environment, training an FL model to detect available spectrum holes, collaborative spectrum sensing by incorporating fusion rules and allocating spectrum resources to the requesting UAVs. For FL-based model training, we propose a proportional weighted federated averaging (pwFedAvg) algorithm that incorporates the received power observed at different UAV locations into the FL model training. The proposed algorithm assigns a higher weight to the UAVs that receive higher signal power.

III. SYSTEM MODEL AND PROBLEM FORMULATION

To model collaborative wideband spectrum sensing and scheduling, we consider a multi-cell wireless network that consists of a set of base stations (BS) denoted by $\mathcal{B}$ ($|\mathcal{B}| = B$), as shown in Fig. 3. In addition, we consider a set of UAVs denoted by $\mathcal{K}$ ($|\mathcal{K}| = K$) in the system. To coordinate the collaborative spectrum sensing, fusion, and scheduling, we assume that each time slot is divided into four consecutive sub-slots: UAV resource request (t_{req}), spectrum sensing (t_s),

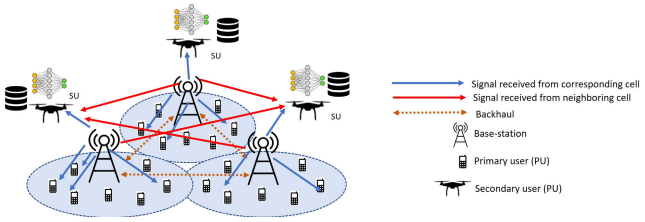


FIGURE 3. A zoomed-in version of dataset generation plane of a multi-cell wireless network with multiple UAVs.

broadcasting to central server (t_b), and channel access (t_a). Specifically, at the beginning of each time slot, the UAVs that require PU resources request the server for resource allocation. In the subsequent sub-slot of sensing (t_s), the UAVs perform spectrum sensing and broadcast the sensed channel information in the following sub-slot (t_b). The central server then applies fusion rules and obtains the spectrum holes. In the access sub-slot (t_a) the central server first assigns the spectrum holes to the requesting UAVs. The UAVs then transmit on the allocated spectrum holes in the access sub-slot (t_a). In an ideal scenario, UAVs would transmit on the allocated spectrum holes immediately. However, in the physical deployment, there would be a delay in forming the packets, adjusting the front-end filters, and transmission by the UAVs. Hence, we assume that in the first time slot, the UAVs will not transmit in the access sub-slot (t_a) but prepare their packets for subsequent time slots. In the subsequent time slots, the UAVs transmit on the spectrum holes that are allocated in the previous time slot. In this paper, we focus on three main stages to develop our proposed framework: (i) FL-based training for wideband spectrum sensing, (ii) collaborative spectrum inference and fusion, and (iii) spectrum scheduling. To coordinate the above three stages, we assume a central server within the UTM ecosystem. Next, we describe these stages.

A. MODEL TRAINING, SPECTRUM INFERENCE, AND SCHEDULING STAGES

1) FL-BASED MODEL TRAINING FOR SPECTRUM SENSING

The UTM system architecture shown in Fig. 1 supports data exchange between multiple UAVs through the USS network. Such a hierarchical architecture makes it feasible to implement FL-based learning algorithms to identify spectrum holes. In this case, we may consider two deployment models within the UTM architecture. One model would be to have a server deployed by each UAV operator where multiple UAVs connected to the operator act as FL clients. The second model would have a server within the USS network that orchestrates multiple UAV operators. Thus, with several UAVs training local models, they exchange model parameters with the central server that is located either at the USS or UAV operator. The central server then aggregates the local model weights according to an aggregation algorithm and transmits the global model weights back to the UAVs to update their local models.

2) COLLABORATIVE SPECTRUM INFERENCE AND FUSION

Due to the highly dynamic environment in which UAVs operate, it may not be feasible for all UAVs to achieve high prediction accuracy across all sub-channels. Therefore, we leverage collaborative spectrum inference by the UAVs and perform fusion at the fusion module within the central server to increase the reliability of spectrum hole detection. In particular, each UAV captures the raw I/Q samples from over-the-air received signals and predicts the availability of spectrum holes across M sub-channels using the FL-trained model. We assume that there is an associated spectrum inference cost for each UAV k involved in sensing at time slot t. The spectrum inference cost is the energy consumed for sensing the spectrum and is proportional to the voltage V_{CC} of the receiver, the system bandwidth W, and the duration allotted for sensing (t_s) [42]. Therefore, it is defined as $SC_{k,m}(t) = t_s V_{CC}^2 W_m$, where W_m is the m-th sub-channel bandwidth. Upon completion of the spectrum inference phase, the UAV k has a predicted spectrum occupancy vector $\widehat{\boldsymbol{h}}_k(t) = [\widehat{h}_{k,1}(t), \ldots, \widehat{h}_{k,M}(t)]$ such that $\widehat{h}_{k,m}(t) = 0$ if the m-th sub-channel is detected vacant at time t, and $\widehat{h}_{k,m}(t) = 1$ otherwise. This problem can be considered as a multi-label classification problem, and we leverage deep neural network (DNN) at each UAV that accepts raw I/Q samples $\boldsymbol{R}_k$ as inputs and outputs the predicted spectrum occupancy vector $\widehat{\boldsymbol{h}}_k(t)$.

The central server receives multiple copies of spectrum holes detected by individual UAVs and applies fusion rules that result in aggregated prediction. In this paper, we use the n-out-of-K fusion rule defined as follows:

$$\widehat{z}_m(t) = \begin{cases} 0, & \text{if } \sum_{k \in \mathcal{K}} \mathbb{1}\{\widehat{h}_{k,m}(t) = 0\} \geq n; \\ 1, & \text{otherwise}, \end{cases} \tag{1}$$

where $\mathbb{1}\{.\}$ is an indicator function. In this case, $\widehat{\boldsymbol{z}}(t) = [\widehat{z}_1(t), \ldots, \widehat{z}_M(t)]$ is the fused prediction of all the M sub-channels at the central server. Note that when $n = 1$, the n-out-of-K rule is equivalent to the "OR" rule, and $n = K$ is the same as the "AND" rule.

3) SPECTRUM SCHEDULING

Based on the aggregated fusion result provided by the fusion module, the central server then allocates sub-channels to the requesting UAVs. The UAVs then transmit data on the sub-channels allocated to them by the server in the next time slot. The transmission energy consumption is denoted by $AC_{k,m}(t)$. The access cost is the energy consumed for data transmission and is defined as $AC_{k,m}(t) = t_a P_{tx}$, where P_{tx} is the transmit power and t_a is the time allotted to transmission. Furthermore, the transmission utility is the amount of data transmitted on the allocated sub-channel, which is defined as

follows:

$$U_{k,m}(t) = t_a W_m \log_2\left(1 + \mathrm{SNR}_{k,m}(t)\right), \tag{2}$$

where $\mathrm{SNR}_{k,m}(t)$ denotes the signal-to-noise ratio for UAV k on sub-channel m.

We highlight that the UAVs transmit on those sub-channels that were detected vacant in the previous time slot $t-1$. Hence, spectrum collision occurs when the previously detected spectrum holes are no longer available at the current time slot t. Note that the true state of sub-channel m at UAV k is denoted by $h_{k,m}(t)$ and is obtained according to Eq. (9). To capture spectrum collisions, we define the spectrum access collision indicator $c_{k,m}(t)$ as follows:

$$c_{k,m}(t) = \begin{cases} 1, & \text{if } h_{k,m}(t) = 0 \text{ and } \widehat{z}_m(t-1) = 0; \\ -1, & \text{if } h_{k,m}(t) \neq 0 \text{ and } \widehat{z}_m(t-1) = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Next, we formulate a joint spectrum sensing and scheduling optimization problem.

B. JOINT SPECTRUM SENSING AND SCHEDULING PROBLEM FORMULATION

Given the presented system model, we now introduce a joint spectrum sensing and scheduling problem to coordinate collaborative spectrum sensing and spectrum scheduling. We cast the problem as the energy efficiency (EE) maximization for the UAVs, where EE is defined as the ratio of system throughput (bits per second) to the total energy consumed for spectrum sensing and transmission (joules). The system throughput is determined by scheduling decisions, with UAVs opportunistically accessing the spectrum resources of the primary network. The energy costs are influenced by the number of UAVs participating in spectrum sensing and the energy consumed during spectrum access. In particular, let $y_{k,m}(t) = 1$ if UAV k is scheduled to use sub-channel m at time t, and $y_{k,m}(t) = 0$ otherwise. Given that the spectrum holes are allocated to the requesting SUs based on the sub-channel availability, we incorporate the sensing and access costs to maximize the overall EE of the system. Therefore, we have:

$$\begin{cases} \max\limits_{\{y_{k,m}(t)\}} & \mathbb{E}\Big\{\sum\limits_{t,k,m} \dfrac{y_{k,m}(t)\, c_{k,m}(t)\, U_{k,m}(t)}{y_{k,m}(t) AC_{k,m}(t) + SC_{k,m}(t)}\Big\} \\ \text{subject to:} & \sum\limits_{m} y_{k,m}(t) \leq 1, \ \forall\, k = 1, 2, 3, \ldots K, \\ & \sum\limits_{k} y_{k,m}(t) \leq 1, \ \forall\, m = 1, 2, 3, \ldots M, \\ & \sum\limits_{k,m} y_{k,m}(t) \leq M - \sum\limits_{m=1}^{M} \widehat{z}_m(t), \\ & y_{k,m}(t) \in \{0, 1\}, \end{cases} \tag{4}$$

where $U_{k,m}(t)$, $SC_{k,m}(t)$, and $AC_{k,m}(t)$ are, respectively, the amount of data transmitted, the sensing cost, and transmission cost by the SU k on sub-band m. The first and second constraints guarantee that each UAV is scheduled to use at most one sub-channel. The expression $M - \sum_{m=1}^{M} \widehat{z}_m(t)$ represents the total number of predicted spectrum holes at time t. Therefore, the third constraint ensures that the total spectrum resources allocated to UAVs across M sub-channels do not exceed the number of detected spectrum holes at time t. We note that the optimization problem in Eq. (4) is a fractional integer programming problem, which is NP-hard in general. Hence, we consider maximizing the numerator alone, which is the total utility $\bar{U}(t)$ of the UAVs over all sub-channels. As a result, the problem becomes an integer programming problem. In this case, the utility would depend on the spectrum usage pattern by the PUs, which is captured by $c_{k,m}(t)$ as well as the channel condition between the BSs and UAVs that determine the amount of transmitted data $U_{k,m}(t)$. To tackle this utility optimization problem, we model the channel occupancy $h_{k,m}(t)$ as a Markov process, enabling us to use an MDP formulation to solve this problem [43] and develop a dynamic spectrum scheduling for the SUs. Further details on how we solve the MDP using RL are discussed in Section V.

Furthermore, as presented in the system model, we note that spectrum usage pattern is captured by the $c_{k,m}(t)$ in Eq. (3), where each UAV is capable of wideband spectrum sensing. Before the spectrum scheduling stage, we have a spectrum sensing stage where each UAV predicts the spectrum holes and the results are fused to obtain $\widehat{z}(t)$ defined in Eq. (1). In this paper, we use a DNN at each UAV to detect spectrum holes, and to train the DNN models, we present an FL-based approach for distributed training of spectrum sensing models.

IV. PROPOSED FL-BASED MODEL TRAINING FOR SPECTRUM SENSING

A. DATASET GENERATION

We assume that each UAV receives signals from more than one BS due to the fact that they operate at higher altitudes, which increases the chances of signal reception from multiple BSs. Furthermore, we assume that the cell bandwidth W is partitioned into M orthogonal sub-channels. Then the total transmitted signal from a BS b across M orthogonal sub-channels at any time t can be represented by the superposition principle as follows:

$$\boldsymbol{s}_b(t) = \sum_{m=1}^{M} I_{b,m}(t)\, \boldsymbol{v}_{b,m}(t), \quad \forall b \in \mathcal{B}, \tag{5}$$

where $I_{b,m}(t) = 1$ if the m-th sub-channel of BS b is occupied at time t, and 0 otherwise. Moreover, $\boldsymbol{v}_{b,m}(t)$ represents the waveform on the m-th sub-channel. As a result, $\boldsymbol{s}_b(t)$ is the transmitted baseband waveform. Each UAV k then receives a wideband signal from multiple BSs in a multi-path propagation environment, which can be expressed as follows:

$$\boldsymbol{r}_k(t) = \sum_{b=1}^{B} \boldsymbol{g}_{k,b}(t) * \boldsymbol{s}_b(t) + \boldsymbol{\eta}_k(t), \quad \forall k \in \mathcal{K}, \tag{6}$$

where $\boldsymbol{g}_{k,b}(t)$ represents the multi-path channel between BS b and UAV k and $\boldsymbol{\eta}_k(t)$ denotes the noise signal observed at

UAV k. Therefore, the signal-to-noise ratio (SNR) observed at UAV k can be written as follows:

$$\text{SNR}_k(t) = \frac{||\sum_{b=1}^{B} \boldsymbol{g}_{k,b}(t) * \boldsymbol{s}_b(t)||^2}{\sigma_k^2(t)}, \quad \forall k \in \mathcal{K}, \tag{7}$$

where $\sigma_k^2(t)$ represents the noise variance observed at UAV k at time t. We use $P_k(t)$ to denote the total power received in UAV k at time t, the numerator term in Eq. (7). The total power received at UAV k is directly proportional to the signal generated as defined in Eq. (5) and the wireless channel as defined in Eq. (6). We will use $P_k(t)$ in proportional weight scaling for FL training.

To train the DNN models for predicting spectrum holes using raw I/Q samples, it has been shown that the characteristics of the wireless signal can be captured by observing only a portion of the signal waveform [9], [15]. Hence, from the received baseband signal $\boldsymbol{r}_k(t)$, we capture J I/Q samples and store them locally. Therefore, the samples from baseband waveform collected at UAV k are represented as $\boldsymbol{R}_k(t)$ given as follows:

$$\boldsymbol{R}_k(t) = \tilde{\boldsymbol{R}}_k(t) + \tilde{\eta}_k(t), \quad \forall k \in \mathcal{K}, \tag{8}$$

where $\tilde{\boldsymbol{R}}_k(t)$ represents the J I/Q samples from the first term in Eq. (6) and the second term represents J complex Gaussian noise samples.

In addition to the I/Q samples, we also need to store the true labels for channel occupancy at each UAV k at time t. The channel occupancy vector $\boldsymbol{h}_k(t)$ is an M-dimensional vector, with each index indicating if a sub-channel m is occupied or free at time t and can be computed as follows:

$$h_{k,m}(t) = \begin{cases} 1, & \sum_{b=1}^{B} I_{b,m}(t) \geq 1; \\ 0, & \text{Otherwise.} \end{cases} \tag{9}$$

Note that $\boldsymbol{h}_k(t)$ observed at time t would be the true label corresponding to the wideband received signal $\boldsymbol{r}_k(t)$. The channel occupancy would remain unchanged for the stored J I/Q samples $\boldsymbol{R}_k(t)$. We store $(\boldsymbol{R}_k(t), \boldsymbol{h}_k(t))$ as an input-output pair that will be used for the training of the FL model. For the sake of simplicity of notation, we represent the input-output pair as $(\boldsymbol{R}_k, \boldsymbol{h}_k)$. In this paper, we assume that both I/Q samples and true labels are available to the UAVs for model training. However, in real-world deployment, the labels must first be collected from the base stations by the central server, where the true labels are then computed according to Eq. (9). The server then broadcasts the true labels back to the UAVs. Note that for each M-dimensional channel occupancy vector $\boldsymbol{h}_k$, the input-output pair is treated as one data sample, and the total I/Q dataset collected at UAV k is denoted as follows:

$$\boldsymbol{D}_k = \{(\boldsymbol{R}_k^1, \boldsymbol{h}_k^1), (\boldsymbol{R}_k^2, \boldsymbol{h}_k^2), \ldots (\boldsymbol{R}_k^{|\boldsymbol{D}_k|}, \boldsymbol{h}_k^{|\boldsymbol{D}_k|})\}, \tag{10}$$

where $|\boldsymbol{D}_k|$ represents the total number of samples in the UAV k. These local datasets are used in FL-based training for spectrum hole detection.

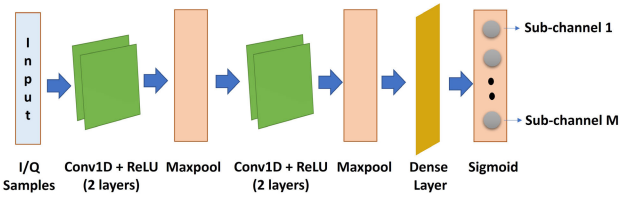


FIGURE 4. Multi-label classification using DNN.

In the FL setting, each UAV k trains a local wideband spectrum sensing model whose parameters are denoted by $\boldsymbol{\omega_k}$. Hence, the primary objective of the local model is to find a mathematical function $f(\boldsymbol{\omega}_k, \boldsymbol{R}_k)$, that maps input I/Q samples $\boldsymbol{R}_k$ to $\boldsymbol{h}_k$, i.e.,

$$f(\boldsymbol{\omega}_k, \boldsymbol{R}_k) : \boldsymbol{R}_k \rightarrow \boldsymbol{h}_k. \tag{11}$$

To this end, using the raw I/Q samples ($\boldsymbol{R}_k$) each UAV k trains a local model that detects vacant sub-channels, such that the local loss function $L_k(\boldsymbol{\omega})$ minimizes the error between the true labels $\boldsymbol{h}_k$ and the predicted labels $\widehat{\boldsymbol{h}_k}$, as defined below:

$$L_k(\boldsymbol{\omega}) \triangleq \frac{1}{|\boldsymbol{D}_k|} \sum_{i=1}^{|\boldsymbol{D}_k|} l(f(\boldsymbol{\omega}_k, \boldsymbol{R}_k^i); \boldsymbol{h}_k^i)), \tag{12}$$

where $l(.)$ is the loss function for computing the prediction loss in the supervised machine learning setting. Furthermore, $f(.)$ represents the predicted label for the sample $(\boldsymbol{R}_k^i, \boldsymbol{h}_k^i)$ and $\boldsymbol{\omega}_k$ represents the local model parameters during training.

For each input sequence $\boldsymbol{R}_k^i$, we intend to obtain an M-dimensional binary vector $\widehat{\boldsymbol{h}_k}^i$ that represents the predicted spectrum holes. This is an instance of a classical multi-label classification problem for which we employ DNN. We next discuss the architecture of the DNN model used for training.

B. DNN MODEL ARCHITECTURE

Previous research works that address classification problems in the wireless domain have shown that filters in the CNNs are highly effective at identifying patterns in the I/Q signals [44], [45]. As mentioned earlier, spectrum hole prediction using I/Q samples is an instance of a multi-label classification problem, and we choose CNNs as our framework of choice. Furthermore, deep networks that use multiple layers of CNNs have been shown to further improve learning performance [46], [47]. Although DNNs that use CNNs have demonstrated significant advantages, they are computationally intensive, especially when handling larger input sizes. However, it has been identified that only a portion of the signal waveform is sufficient to identify the spectrum holes [9], [15]. Hence, in this work, we only capture J I/Q samples from the waveform, which significantly decreases computational complexity. By reducing the input size, we obtain a lightweight model suitable for deployment on UAVs, which typically have limited onboard computational resources.

The DNN architecture considered is shown in Fig. 4. The model accepts raw I/Q samples as input, specifically a tensor of size $(J, 2)$, where J is the number of I/Q samples, defined

in Eq. (8). The I/Q samples are then processed by two one-dimensional (1D) convolutional layers (Conv1D) followed by a 1D maximum pooling layer (MaxPool1D), where each MaxPool1D reduces the output dimension by half. When the input size is small, a network with many MaxPool1D layers would reduce the tensor size and minimize the depth of the network very quickly. Hence, we choose one MaxPool1D layer for two Conv1D layers in our architecture. This layered pattern is repeated twice, and one dense layer is followed by a sigmoid layer at the end. By using a sigmoid function as the activation function with binary cross-entropy as the loss function at the output, each output neuron is independent of the other, which creates a multi-label classifier.

C. CHANNEL-AWARE FL AGGREGATION METHOD

Given the system model, we now introduce a framework for wideband spectrum sensing where multiple UAVs collaboratively participate in the FL. In such a distributed learning environment, we aim to learn a global statistical model at the central server. Given that each UAV k trains a local model to identify the spectrum holes by minimizing the local loss function $L_k(\boldsymbol{\omega})$, in the context of FL, we would like to minimize the aggregated global loss function $L(\boldsymbol{\omega})$, as follows:

$$\min_{\boldsymbol{\omega}} \left\{ L(\boldsymbol{\omega}) \triangleq \sum_{k=1}^{K} \frac{|\boldsymbol{D}_k|}{D} L_k(\boldsymbol{\omega}) \right\}, \tag{13}$$

where $D = \sum_{k=1}^{K} |\boldsymbol{D}_k|$ is the total size of data samples across the UAVs.

To solve the global loss function in Eq. (13), the authors in [48] proposed FedAvg, which is an iterative aggregation algorithm by which the global model aggregates the local model weights and redistributes the global model weights to the local models. In particular, the central server first broadcasts the latest model weights, $\boldsymbol{\omega}^t$, to all the UAVs. Second, each UAV k adjusts its weights as $\boldsymbol{\omega}_k^t = \boldsymbol{\omega}^t$ and performs E local updates as follows:

$$\boldsymbol{\omega}_k^{t+i+1} = \boldsymbol{\omega}_k^{t+i} - \gamma^{t+i} \nabla L_k(\boldsymbol{\omega}_k^{t+i}; \boldsymbol{\xi}_k^{t+i}), i = 0, \ldots, E-1, \tag{14}$$

where γ^{t+i} is the learning rate of the local model and $\boldsymbol{\xi}_k^{t+i}$ is the random batch of samples from the local data $\boldsymbol{D}_k$. Then, the server aggregates the local model weights to produce new global model weights as follows:

$$\boldsymbol{\omega}^{t+E} = \sum_{k=1}^{K} p_k^{t+E} \boldsymbol{\omega}_k^{t+E}, \tag{15}$$

where $\sum_{k=1}^{K} p_k^{t+E} = 1$, for any t and E. The design of the aggregation weights p_k^{t+E} in Eq. (15) has been investigated in recent studies. For instance, the authors in [38] propose optimizing federated averaging over fading channels that allocates non-uniform weights to different clients based on the observed instantaneous channel state information at the receiver (CSIR). Similarly, [37] and [49] present federated learning approaches over noisy channels that allocate varying weights, which are expressed in terms of the SNR values observed between the clients and the server in the model training plane. The authors in [35], [39], [40], [41] consider device heterogeneity and assign weights to clients based on their quantization errors.

In our considered multi-cell environment, the signal received at different UAV locations experiences different channel conditions, and the signal power received at different locations varies significantly. Hence, by assigning equal scaling weights for the local model gradients, the performance metrics at UAV locations with strong signals deteriorate. To compensate for this effect and improve performance at locations that receive better signal power, we propose a proportional weight scaling aggregation method for FL (pwFedAvg) that assigns weights to UAVs proportional to their received signal power, which is similar to [38] that assumes a wireless channel in the model training plane. Moreover, as mentioned earlier, we only capture a portion of I/Q samples. The CNNs identify the patterns in the I/Q signals by training a model, whose weights would be proportional to the input signal. This is in line with the derivation of binary cross entropy loss gradient, where the gradients are directly proportional to the input [50], [51].

To this end, we propose proportional weights as follows:

$$p_k^{t+E} = \sum_{k=1}^{K} \frac{\alpha_k^{t+E}}{\alpha^{t+E}}, \tag{16}$$

where $\alpha_k^{t+E} = \sqrt{\bar{P}_k}$ and $\alpha^{t+E} = \sum_{k=1}^{K} \sqrt{\bar{P}_k}$. Here, $\bar{P}_k$ represents the average received signal power at UAV k for the batch of samples $\boldsymbol{\xi}_k^{t+i}$ up to E epochs. The overall process of FL-based model training using the pwFedAvg approach is outlined in Algorithm 1. Next, we discuss convergence analysis.

Algorithm 1 Channel-Aware FL-Based Training

1: Initialize the global model parameters $\boldsymbol{\omega}$ and local model $\boldsymbol{\omega_k}, \forall k \in \mathcal{K}$; T : Communication rounds.
2: **for** t in T **do**
3: **for** UAV k in $\mathcal{K}$ **do**
4: Choose a batch of I/Q samples $\boldsymbol{\xi}_k^t \subseteq \boldsymbol{D}_k$.
5: Train the local model for E epochs per Eq. (14).
6: **end for**
7: Broadcast the weights $\boldsymbol{\omega}_k^{t+E}$ to the central server.
8: Aggregate local model weights at the server per Eq. (15).
9: Update local models using the global model weights, i.e., $\boldsymbol{\omega}_k^{t+1} = \boldsymbol{\omega}^{t+1}$.
10: **end for**

D. CONVERGENCE ANALYSIS

In this section, we provide the convergence analysis of the proposed pwFedAvg algorithm. To this end, we first introduce the following assumptions.

Assumption 1: The loss function $L_k(.)$ is L-smooth, i.e., for all $\boldsymbol{u}$ and $\boldsymbol{v}$,

$$L_k(\boldsymbol{u}) - L_k(\boldsymbol{v}) \leq (\boldsymbol{u}-\boldsymbol{v})^T \nabla L_k(\boldsymbol{v}) + \frac{L}{2}||\boldsymbol{u}-\boldsymbol{v}||_2^2. \quad (17)$$

Assumption 2: For each k, $L_k(.)$ is β-strongly convex, i.e., for all $\boldsymbol{u}$ and $\boldsymbol{v}$,

$$L_k(\boldsymbol{u}) - L_k(\boldsymbol{v}) \geq (\boldsymbol{u}-\boldsymbol{v})^T \nabla L_k(\boldsymbol{v}) + \frac{\beta}{2}||\boldsymbol{u}-\boldsymbol{v}||_2^2. \quad (18)$$

Assumption 3: Let $\boldsymbol{\xi}_k$ be the data samples chosen from $\boldsymbol{D}_k$. The variance of the stochastic gradients for each UAV k is bounded i.e,

$$\mathbb{E}\left[||\nabla L_k(\boldsymbol{\omega}_k;\boldsymbol{\xi}_k) - \nabla L_k(\boldsymbol{\omega}_k)||_2^2\right] \leq \rho_k^2 \ \forall\, k \in \mathcal{K}. \quad (19)$$

Assumption 4: The expected squared norm of the stochastic gradients for each UAV k is bounded as follows:

$$\mathbb{E}\left[||\nabla L_k(\boldsymbol{\omega}_k;\boldsymbol{\xi}_k)||_2^2\right] \leq G^2 \ \forall\, k \in \mathcal{K}. \quad (20)$$

The expectations used in the analysis are taken over $\boldsymbol{\xi}_k$ unless otherwise stated where $\boldsymbol{\xi}_k$ are the data samples drawn from $\boldsymbol{D}_k$. Next, similar to [49], [52], we define two virtual sequences to denote the aggregated full gradient and stochastic gradient respectively, as follows:

$$\bar{\boldsymbol{a}}^t = \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \nabla L_k(\boldsymbol{\omega}_k^t); \quad \boldsymbol{a}^t = \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \nabla L_k(\boldsymbol{\omega}_k^t;\boldsymbol{\xi}_k^t). \quad (21)$$

We also assume that $\mathbb{E}[\boldsymbol{a}^t] = \bar{\boldsymbol{a}}^t$. Given these assumptions, we have the following lemmas.

Lemma 1: Let $\boldsymbol{\omega}^* = [\omega_1^*, \omega_2^*, \ldots, \omega_d^*]$ be the weights of optimal global model, and $\boldsymbol{\omega}_k^* = [\omega_{k,1}^*, \omega_{k,2}^*, \ldots, \omega_{k,d}^*]$ be the weights of optimal local model of UAV k. Here, d represents the dimensions of the model weights. Then for each UAV k, the upper bound of the gap between the optimal global and local models can be shown as,

$$L_k(\boldsymbol{\omega}^*) - L_k(\boldsymbol{\omega}_k^*) \leq \tau, \quad (22)$$

where $\tau = \max_k\{\frac{Ld}{2}(\max_i\{|\omega_i^* - \omega_{k,i}^*|\})^2\}$.

Lemma 2: The aggregated gradient is upper bounded as follows:

$$\mathbb{E}\left(||\boldsymbol{a}^t - \bar{\boldsymbol{a}}^t||_2^2\right) \leq \sum_{k=1}^{K}\left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \rho_k^2. \quad (23)$$

Lemma 3: Let the constants κ and γ^t satisfy $\frac{1}{\kappa} \leq \gamma^t$. Then, we can show that:

$$\mathbb{E}\left[||\boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^*||_2^2\right] \leq (1-\beta\gamma^t)||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2 + (\gamma^t)^2\zeta^t, \quad (24)$$

where $\zeta^t = 2\kappa\tau + 4(E-1)^2G^2 + \sum_{k=1}^{K}\left(\frac{\alpha_k^t}{\alpha^t}\right)^2\left[\rho_k^2 - 2L\tau\right]$.

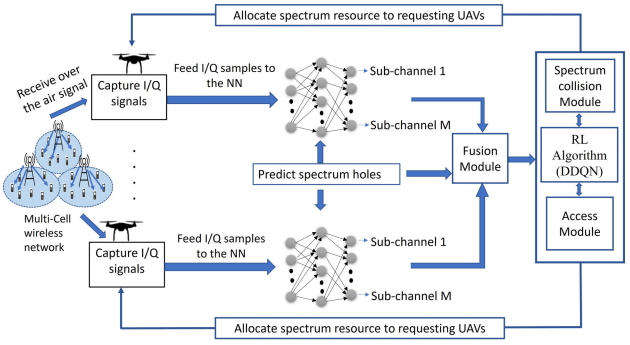


FIGURE 5. Joint spectrum inference and spectrum scheduling.

Theorem 1: Given that $\kappa \leq \gamma^t = \frac{1}{\beta t + L}$, the optimality gap for the proposed pwFedAvg satisfies the following:

$$\mathbb{E}[L(\boldsymbol{\omega})^T] - L^* \leq \frac{L}{\beta T + 2L}\left[\frac{2\zeta}{\beta} + L\,\mathbb{E}||\boldsymbol{\omega}^0 - \boldsymbol{\omega}^*||_2^2\right], \quad (25)$$

where $\zeta = \max_t\{\zeta^t\}$ and ζ^t is as defined in Lemma 3. Therefore, we show that the convergence of our proposed method is $\mathcal{O}(\frac{1}{T})$. Assuming that each UAV performs E local epochs during training, an additional term emerges in Eq. (24) that directly influences the optimality gap presented in Eq. (25). Furthermore, our convergence analysis is based on the assumption of independent and identically distributed (i.i.d.) data. However, when considering non-i.i.d. data, an additional term related to data heterogeneity will need to be taken into account in Eq. (25). The convergence analysis for non-i.i.d. data can be found in [52] and [53]. All of the proofs are presented in Section IX.

V. DYNAMIC SPECTRUM SCHEDULING USING RL

Once the DNN models are trained using our proposed pwFedAvg, they output their spectrum hole predictions, which are then fused at the fusion module, as described in Section III. The identified spectrum holes will be allocated to requesting UAVs. The integrated system model of collaborative spectrum sensing and scheduling is shown in Fig. 5, with the overall algorithm described in Algorithm 2.

For spectrum scheduling, we note that the optimization problem in Eq. (4) is a fractional integer programming problem, which is NP-hard in general. If we consider maximizing the numerator alone, which is the total utility $\bar{U}(t)$ of the UAVs over all sub-channels, the problem will become an integer programming problem. In this case, the utility would depend on the spectrum usage pattern by the PUs, which is captured by $c_{k,m}(t)$ as well as the channel condition between the BSs and UAVs that determine the amounts of transmitted data $U_{k,m}(t)$. To tackle this utility optimization problem, we model the channel occupancy $\boldsymbol{h}(t)$ as a Markov process, enabling us to use an MDP formulation to solve this problem [43] and develop a dynamic spectrum scheduling for the SUs.

Algorithm 2 Collaborative Spectrum Sensing and Scheduling

Phase 1 – Spectrum Sensing and Broadcasting

1: **for** each UAV in $\mathcal{K}$ **do**
2: Capture I/Q samples from over-the-air signal.
3: Feed I/Q samples to the pre-trained ML model that predicts the spectrum holes $\widehat{\boldsymbol{h}}$.
4: Broadcast the individual spectrum hole observations $\widehat{\boldsymbol{h}}(t) \in \{0, 1\}^{1\times M}$ to the central server.
5: **end for**

Phase 2 – Spectrum Fusion and Scheduling

6: Apply fusion rule in Eq. (1) to predict spectrum holes $\widehat{z}(t)$.
7: Allocate a single spectrum hole to each requesting UAV using a pre-trained RL algorithm, $y_{k,m}(t)$, such that the constraints in Eq. (4) are satisfied.
8: UAVs are scheduled to transmit on the sub-channel allocated in the previously allocated time slot.
9: Given the spectrum allocation $y_{k,m}(t)$ and spectrum access collision indicator $c_{k,m}(t)$, the total utility $\widehat{U}(t)$ can be computed using Eq. (4).

As we assume that there exist M sub-channels in the system, each sub-channel can be modeled as an independent two-state Markov chain. The transition probability function $\boldsymbol{P}$ can then be viewed as a set of transition probability matrices $\{\boldsymbol{P}_m\}$ for each sub-channel that captures the randomness in the assumed multi-user multi-channel environment. Therefore, we formulate the total utility of the SUs as a traditional MDP governed by the tuple $(\mathcal{S}, \mathcal{A}, \{\boldsymbol{P}_m\}, \mathrm{U}, \gamma)$, consisting of the set of states $\mathcal{S}$, set of actions $\mathcal{A}$, a transition probability function $\{\boldsymbol{P}_m\}$, a reward function U, and a discount factor γ. To solve an MDP using RL, an agent learns to make decisions in an uncertain environment by maximizing a cumulative reward over a sequence of actions. Specifically, the agent interacts with an environment by taking actions that transition the system from one state to another, and the agent receives a reward that is commensurate with the merit of the action. The discount factor determines the relative importance of immediate and future rewards.

A. DDQN-BASED SPECTRUM ALLOCATION

One of the most popular RL methods is Q-learning [43]. The classical Q-learning is table-based, i.e. the values of the Q-function are stored in a table of size $|\mathcal{S}|\times|\mathcal{A}|$. However, when the size of the state and action spaces is large, the complexity of tabular Q-learning becomes cumbersome. For example, with $M = 16$ sub-channels, the Q-table will be of size $65,537 \times 17$. To address the complexity issue, we adopt the deep Q-learning approach in [54] to approximate the Q-function by a neural network $Q_{\boldsymbol{\theta}}$ called DDQN and train its weights $\boldsymbol{\theta}$ using experience replay. As the name suggests, we have two networks when using DDQN where $Q_{\boldsymbol{\theta}}$ is called the primary network, and $Q'_{\boldsymbol{\theta}}$ is called the target network and the weights of the target network are updated periodically. In the original DDQN, the weights of the target network are directly copied from the primary network every few episodes. In DDQN-soft, the target networks are updated using POLYAK averaging to smoothly update the weights ("soft-update") [54].

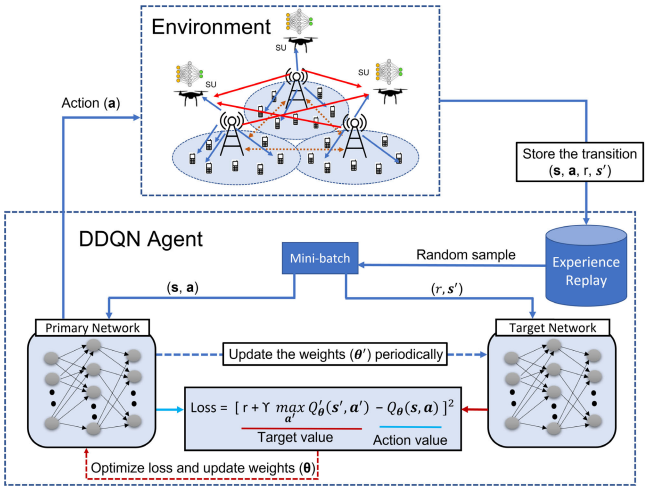


FIGURE 6. DDQN for spectrum allocation.

The input to the DDQN agent is a state $\mathbf{s}$ of size $1 \times M$. The output of the network is a vector of size $1 \times (M + 1)$ that contains the values of the Q-function with respect to state $\mathbf{s}$ and each of the $M + 1$ actions. In all hidden layers, we use the rectified linear unit (ReLU) as an activation function. Given the neural network's input-output dimensions, the overall DDQN architecture and its interaction with the environment are shown in Fig. 6. As shown, the major components are the primary network, target network, experience replay, and the interaction with the environment to select an action.

To train the DDQN agent, the experiences are initially stored in the memory using ϵ-greedy policy, that is, for a state s_t, an action a_t is taken randomly with probability ϵ_t, or taken greedily with probability 1-ϵ_t from the current state of the DDQN network. Then, when we have sufficient samples in the memory a mini-batch of $\boldsymbol{X}$ experiences $\{(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i)\}_i \in \boldsymbol{X}_t$ is randomly sampled from memory for every time step t to train the neural networks. Here, $\boldsymbol{X}_t$ is the set of experiences currently available in the memory. Based on the selected mini-batch, we compute and update the weights $\boldsymbol{\theta}$ of the primary network $Q_{\boldsymbol{\theta}}$ that minimize the loss function $L_t(\boldsymbol{\theta})$. Fig. 6 captures the overall DDQN architecture and the interaction of the agents with the environment [43], [54].

VI. I/Q DATASET GENERATION

Utilizing data-driven machine-learning techniques for wide-band spectrum sensing requires substantial amounts of spectrum data. While obtaining raw I/Q signals over the air

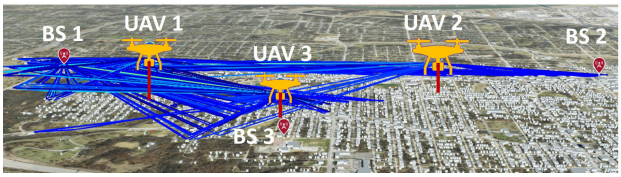


FIGURE 7. Ray-tracing simulation setup used for dataset generation. The plot illustrates the received signal paths at UAV location 1 from all three base stations.

using physical hardware is the ideal scenario, the complexity of coordinating multiple UAVs in a specific environment for collaborative sensing poses significant challenges in achieving this objective. Therefore, we resort to MATLAB's LTE toolbox to create the I/Q samples and employ ray-tracing methods to emulate the channel for generating synthetic datasets that closely mimic the data collection process through experimentation. The entire process of generating synthetic datasets is outlined below.

A. DATASET GENERATION METHODOLOGY

As shown in Fig. 7, we assume a multi-cell environment consisting of three neighboring cells with base stations at the center of the cells. Without loss of generality, we simulate one specific LTE band in the Kansas City area and obtain the location of the base stations from cellmapper [55], an open crowd-sourced cellular tower and coverage mapping service. Furthermore, we assume there are three UAVs in the network operating at an altitude of 90 meters. In this scenario, the base stations act as the transmitter sites, and the UAV locations as the receiver sites that collect the I/Q samples for wide-band spectrum sensing.

Another important aspect of any wireless network is the wireless channel modeling. We use ray-tracing methods to incorporate the channel between the BS and the UAV. We incorporate both reflection and diffraction settings in ray-tracing to simulate a near real-world environment. This is in contrast to using channel models, which consider probabilistic channel models for line-of-sight (LoS) and non-LoS channel conditions. Since we use ray-tracing, we have the flexibility to incorporate different aspects of the environment like buildings and vegetation, permittivity, and permeability of the materials, which further enhances the channel model.

To mimic the real-world scenario for conducting ray-tracing experiments, we use OpenStreetMap, which is a free and open geographical database [56]. The evaluation area is a 3 km $\times$ 3 km area with buildings and vegetation. We utilize MATLAB's ray tracer to emulate the wireless channel between considered UAVs and base station locations. The ray-tracing simulation setup is outlined in Table 1.

It is essential to note that this setup can be seamlessly adapted to accommodate varying numbers of LTE cells and UAVs as long as we can obtain the 3D environment and load it into MATLAB. The MATLAB's ray-tracing toolbox effectively emulates the channel. As illustrated in Fig. 7,

TABLE 1. Ray-tracing simulation setup.

Parameter	Description
Location	Kansas City
Area	3 km x 3 km
Frequency	1980 MHz
Number of base stations	3
Number of UAVs	3
UAV Altitude	90 m
Max. Number of Reflections	5
Max. Number of Diffractions	2

datasets are generated for three UAV locations, with the UAVs positioned randomly and spatially separated. The three UAV locations can be interpreted either as three independent UAVs simultaneously capturing data or as a single UAV collecting data at three different points along its trajectory. However, in the latter scenario, with only one UAV involved, it is not possible to explore FL. In the former scenario, we assume the UAVs are stationary and hover in a fixed position. In general, this simulation can be extended to incorporate UAV flight trajectories by running additional ray-tracing experiments for each UAV way-point location in the UAV trajectory.

Next, we utilize MATLAB's LTE Toolbox to generate the LTE waveform to extract the I/Q samples. For generating the LTE waveform, we assume that the entire cell bandwidth of 10 MHz (50 resource blocks) is split into 16 orthogonal sub-channels, each of size 3 resource blocks. Typically, a base station has the flexibility to assign either a single sub-channel or multiple sub-channels to a PU for transmitting user-specific data on the downlink shared channel. Additionally, various multiple access techniques can be employed to transmit data to different PUs in different time slots. However, during our dataset generation process, we do not consider primary user locations and how the base station allocates user-specific data to different PUs. At any given point in time, we take a snapshot of the entire cell bandwidth and identify which spectrum bands are occupied. Furthermore, when creating the downlink waveform, we omit the generation of user-specific reference signals to avoid mixing user-specific data with broadcast channels. Instead, we identify the appropriate indices and embed the LTE data samples into the downlink shared channel to generate the LTE waveform. We would like to emphasize that other multi-cell networks and wireless technologies can also be considered instead of LTE, but the evaluation scenarios and dataset generation steps need to be modified accordingly. In this paper, we utilize MATLAB's LTE Toolbox to extract the I/Q samples received by UAVs in a multi-cell environment, specifically for wide-band spectrum sensing purposes.

B. MODELLING THE CHANNEL OCCUPANCY

In our assumed scenario, each cell bandwidth is divided into 16 sub-channels such that a binary flag 1 indicates the

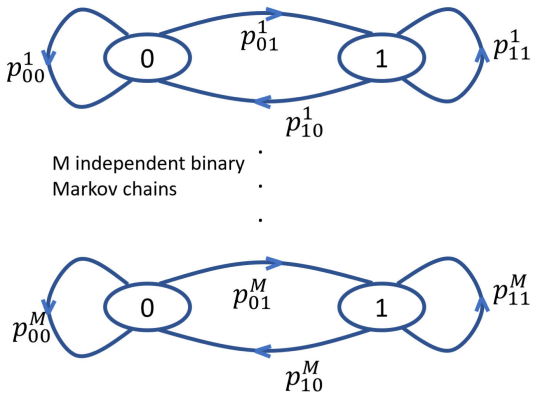


FIGURE 8. **M independent Binary Markov chains. In our dataset generation, we set M = 16.**

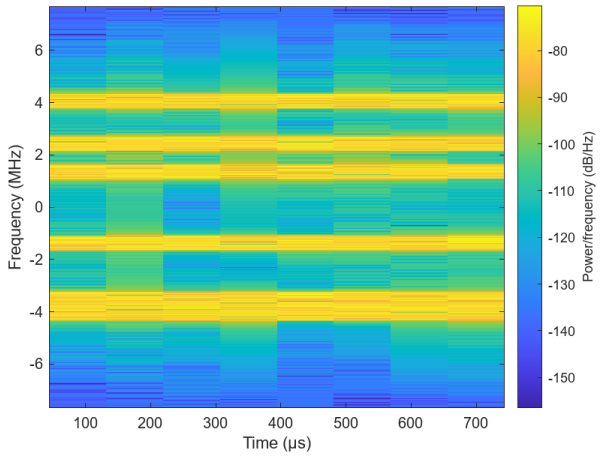


FIGURE 9. **Spectrogram for a transmission on 6 sub-channels out of 16 sub-channels.**

sub-channel is allocated and 0 represents the sub-channel is not allocated. Hence, each 16-bit binary combination serves as a distinct true label for the channel occupancy. As a result, the base station has the capability to generate 2^{16} unique labels, spanning from no sub-channel allocation to a fully busy cell site. For instance, in Fig. 9 we show the spectrogram of one channel realization, where 6 sub-channels are occupied out of 16 sub-channels. Furthermore, we model the temporal dynamics of each sub-channel using a binary Markov chain, as shown in Fig. 8. Thus, the channel occupancy for each sub-channel m evolves according to a transition probability matrix $\boldsymbol{P}_m$. In this paper, we consider different transition probabilities for each sub-channel. Thus, the overall transition probabilities across M sub-channels are denoted as follows:

$$\boldsymbol{P} = \left\{ \begin{bmatrix} p_{00}^1 & p_{01}^1 \\ p_{10}^1 & p_{11}^1 \end{bmatrix}, \cdots\cdots, \begin{bmatrix} p_{00}^M & p_{01}^M \\ p_{10}^M & p_{11}^M \end{bmatrix} \right\}. \tag{26}$$

Further, we assume that all SUs are capable of receiving the waveform from all the base stations, whose channel is modeled by ray-tracing. In addition to the reflected paths received from the corresponding base station in which the

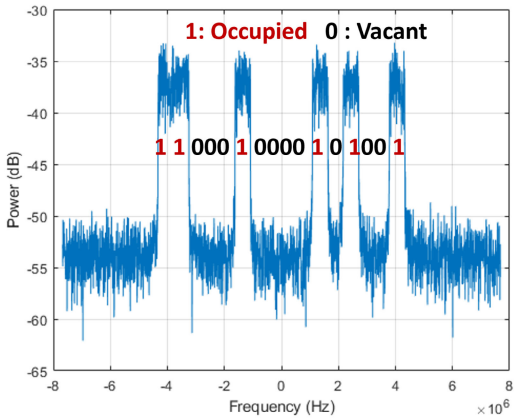


FIGURE 10. **Power spectrum observed at UAV 1 for transmission on 6 sub-channels out of 16 sub-channels, with channel impairments and for snr=−10dB; Vacant sub-channels are indicated as 0 and occupied sub-channels by 1.**

UAV is present, we also receive the waveform from the neighboring base stations as shown in Fig. 7. The received signal $\boldsymbol{r}_k(t)$ at each UAV k can be written as a superposition of wideband signals received from all base stations as shown in Eq. (6).

In this work, we assume i.i.d. datasets across the UAVs. In particular, as shown in Eq. (6), while each UAV observes a different wireless channel from a base station, it still receives signals from all base stations. As a result, each UAV captures the same quantity of I/Q samples with the same distribution of usage patterns (i.e., labels) across the spectrum sub-channels. However, in real-world deployments, signal loss at certain locations may lead to different UAVs capturing varying amounts of data with different spectrum usage patterns, resulting in non-i.i.d. data distributions across different UAVs.

We vary the noise variance $\sigma_k^2(t)$ at UAV k such that the effective SNR varies from -10 dB to 20 dB in steps of 10 dB. For instance, in Fig. 10 we show the power spectrum of the received signal at UAV location 1. Ideally, we would like to capture the whole LTE frame corresponding to the 10 MHz LTE waveform. However, we only capture 32 I/Q samples that provide a good trade-off between the computational complexity and the performance. In this context, for each SNR, we collect approximately 6.8 million I/Q samples. Considering all the SNR levels and the UAV locations, the total generated dataset is more than 80 million I/Q samples, which will be publicly released along with all source codes. The generation of such large-scale spectrum datasets for dynamic UAV environments enables us to evaluate the proposed data-driven collaborative wideband spectrum sensing and sharing, as described next.

VII. NUMERICAL RESULTS

In this section, we first present our target performance metrics, followed by a discussion of the results of spectrum

sensing for different ML configurations. Next, we present the results of collaborative spectrum inference and fusion followed by spectrum access using RL.

Performance metrics: As mentioned earlier, detecting spectrum holes aligns with the framework of a classical multi-label classification problem, where each sub-channel represents a label. We utilize Precision, Recall, and F1-score as metrics to evaluate the classifier's performance for each sub-channel by constructing a confusion matrix. Although we can calculate these performance metrics for each sub-channel individually, it would be advantageous to have an average performance assessment across all 16 sub-channels [57]. In this paper, we consider the micro-averages for Precision, Recall, and F1-score to concretely capture the wideband spectrum sensing performance across the 16 sub-channels as follows:

$$\text{Precision} = \frac{\sum_{m=1}^{M} \text{TP(m)}}{\sum_{m=1}^{M} \text{TP(m)} + \text{FP(m)}}, \tag{27}$$

$$\text{Recall} = \frac{\sum_{m=1}^{M} \text{TP(m)}}{\sum_{m=1}^{M} \text{TP(m)} + \text{FN(m)}}, \tag{28}$$

$$\text{F1-score} = \frac{2(\text{Precision. Recall})}{\text{Precision} + \text{Recall}}, \tag{29}$$

where TP, FN, and FP account for the number of true positives, false negatives, and false positives, respectively. A FP occurs when a sub-channel is declared a spectrum hole while the sub-channel is currently occupied. If such a falsely detected spectrum hole is assigned to the UAVs, we introduce unwanted interference into the PU network. On the other hand, FN misses out on the vacant spectrum hole and declares the sub-channel as busy. Hence, we ideally want lower FP and FN which implies we obtain higher Precision and Recall metrics. F1-score as defined in Eq. (29) is the geometric mean of Precision and Recall.

A. SPECTRUM SENSING WITH DISTRIBUTED UAVs

1) MODEL TRAINING METHODS

As previously stated, the goal is to achieve wideband spectrum sensing to identify spectrum holes from the given I/Q samples as inputs to the ML model. In this context, we explore three model training configurations: centralized learning (CL), local learning (LL), and federated learning (FL), to train the wideband spectrum sensing model. In each of these configurations, we use 70% of the dataset to train the model and 30% for spectrum inference (testing) purposes.

Centralized Learning (CL) is a technique in which it is assumed that all the data collected at different locations are aggregated at one central server and are readily available to train the ML model. At the end of the training process, there will be one central model that can be tested at each UAV location.

Local Learning (LL) is an ML technique in which each UAV trains a model with its own local data, without sharing the dataset or model parameters with a central server or other UAVs. Hence, LL characterizes the performance of the model at a particular location. At the end of the training process, each UAV will have its own locally trained model.

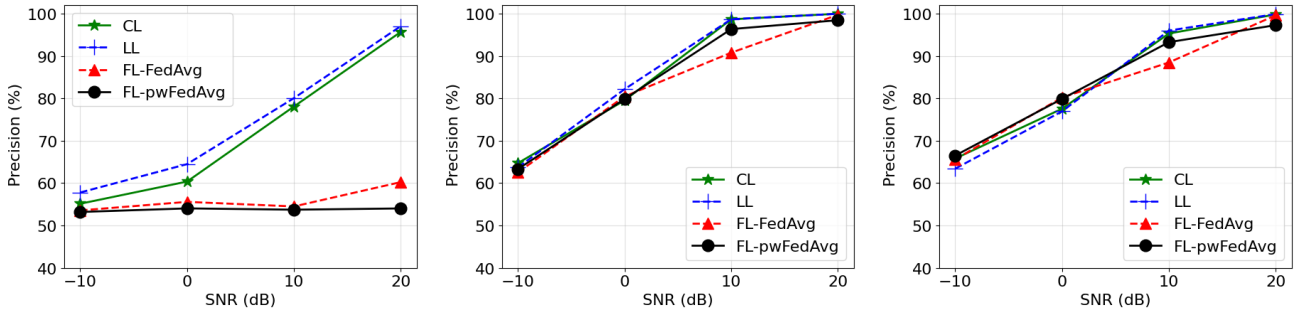
Federated Learning achieves a trade-off between LL and CL, as it does not require aggregating the datasets in a central location; instead, the local model weights are transferred to the central server for aggregation, and in return, the local models receive aggregated global weights. As such, the training process is similar to LL except that the local model weights are updated with the computed global weights iteratively. To investigate FL performance, we implement the FedAvg algorithm [48] as a baseline alongside our proposed pwFedAvg described in Algorithm 1. At the end of the training process, all UAVs will have the same global model. In addition to FL-based methods, the dynamic nature of UAVs suggests the potential utility of online learning methods (see, for example, [58]) for spectrum sensing. However, we observed that as the data collected at different UAV locations varied significantly, the model experienced catastrophic forgetting. This caused substantial changes in the model weights and made the convergence of online learning more challenging. Next, we compare the results obtained for each configuration.

2) TRADE-OFF OF CL, LL, AND FL METHODS

The trained models are loaded on the UAVs for testing purposes, and the performance metrics are computed at different UAV locations for different SNRs. We note that each UAV observes a different wireless channel, and thus the received signal power is different at different locations. In Fig. 11, Fig. 12, and Fig. 13, we compare the Precision, Recall, and F1-score obtained at each UAV location for all the configurations. To use the CL method, we need to accumulate all the datasets in one place and train one central model. Hence, the CL model performs well in different locations, as shown in Fig. 11, Fig. 12, and Fig. 13. However, a major drawback is the need to aggregate all datasets at a single location, which may not always be feasible due to resource constraints and security concerns. Therefore, while CL might be suitable for small datasets, it becomes impractical for larger datasets.

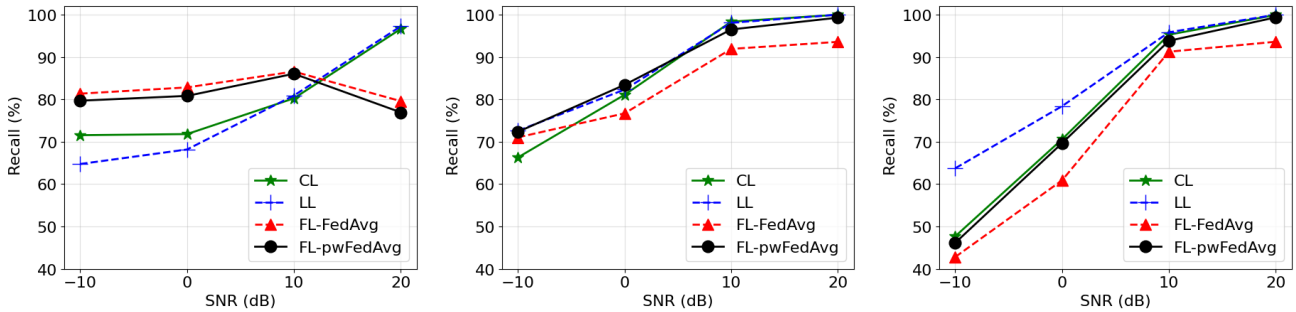
On the other hand, in LL we train a separate model for each UAV with their location-specific data. While LL models perform comparably to CL models and address CL's limitations, the LL models cannot be generalized to other locations. For example, in Fig. 14, when the local model trained at the UAV location 1 is tested at locations 2 and 3 (denoted by LL-1-2 and LL-1-3), the performance metrics are significantly lower than the performance metrics obtained by testing the local model tested at location 1 itself (denoted by LL-1-1). This is one of the key observations that led us to explore FL which combines the advantages of both LL and CL to obtain a more generalized global model, without the need to aggregate all the datasets in one central location.

Comparing the performance of the model training methods across different locations, we find that the performance metrics at UAV locations 2 and 3 are superior to those at location



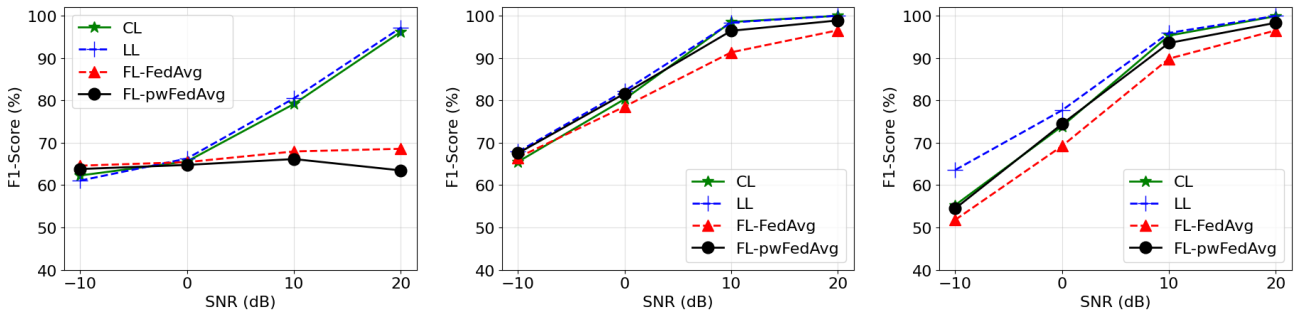
(a) Location 1. (b) Location 2. (c) Location 3.

FIGURE 11. Comparison of Precision obtained at UAV locations 1, 2, 3 for CL, LL, FL-FedAvg, and FL-pwFedAvg.



(a) Location 1. (b) Location 2. (c) Location 3.

FIGURE 12. Comparison of Recall obtained at UAV locations 1, 2, 3 for CL, LL, FL-FedAvg, and FL-pwFedAvg.



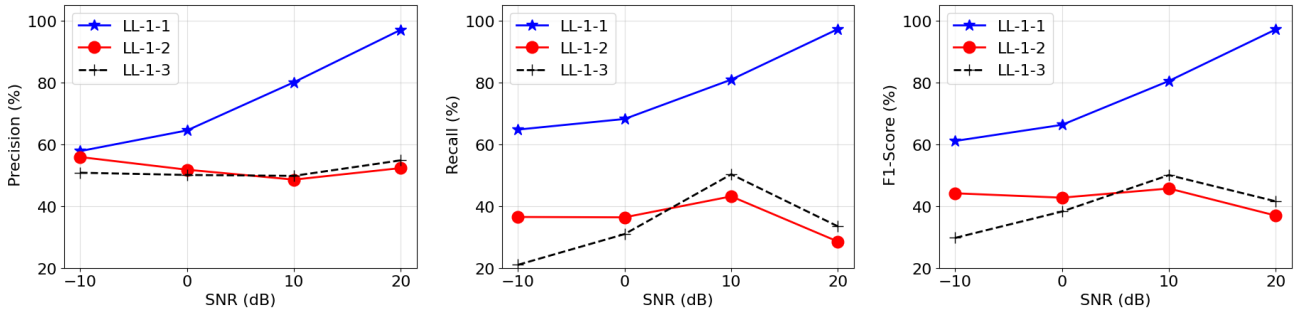
(a) Location 1. (b) Location 2. (c) Location 3.

FIGURE 13. Comparison of F1-score obtained at UAV locations 1, 2, 3 for CL, LL, FL-FedAvg, and FL-pwFedAvg.

TABLE 2. Comparison of F1-score results with and without fusion for CL, LL, FL-FedAvg, and FL-pwFedAvg at UAV location 1.

SNR	-10 dB		0 dB		10 dB		20 dB	
	w/o Fusion	Fusion	w/o Fusion	Fusion	w/o Fusion	Fusion	w/o Fusion	Fusion
Centralized Learning (CL)	62.29%	65.07%	65.60%	80.23%	79.15%	97.86%	96.17%	99.96%
Local Learning (LL)	61.08%	67.74%	66.28%	82.23%	80.49%	98.08%	97.15%	99.98%
FL-FedAvg	64.60%	66.79%	65.43%	79.23%	67.96%	91.52%	68.79%	97.23%
FL-pwFedAvg	63.80%	67.46%	64.77%	81.37%	66.16%	95.45%	63.48%	98.08%

1. For example, as shown in Fig. 11a, precision saturates around 96% at the SNR of 20 dB for CL and LL at location 1. In contrast, for locations 2 and 3, precision improves with increasing SNR, reaching 99.5% at 20 dB. A similar trend is observed for Recall and F1-score, as illustrated in Fig. 12a, and Fig. 13a.



(a) Precision. (b) Recall. (c) F1-score.

FIGURE 14. Comparison of performance metrics obtained for testing Local Learning (LL) model 1 at different UAV locations. Here, LL-1-1 refers to LL model 1 (trained at location 1) tested at location 1, LL-1-2 refers to LL model 1 tested at location 2, and LL-1-3 refers to LL model 1 tested at location 3. From the results, we observe that the local model does not generalize well to new locations.

3) FedAvg VERSUS pwFedAvg PERFORMANCE

To investigate the FL performance, we first implemented the FedAvg algorithm [48], assuming each FL client performs $E = 10$ local updates. From the results in Figs. 11- 13, we note that FedAvg achieves good performance only for the UAV locations 2 and 3. Given the datasets collected at different UAV locations, the performance metrics of FL, compared to CL and LL, are significantly impacted by the UAV(s) that perform worst due to the model aggregating step. This is because FedAvg scales the weights of all local models equally. To reduce the impact of UAV locations with poor performance, our proposed pwFedAvg algorithm scales the weights of local models according to the received signal power according to Eq. (15).

As shown in Fig. 11, Fig. 12, and Fig. 13, we note that the proposed proportional weighting scheme improves F1-score at locations 2 and 3. As shown in Fig. 13, the F1-score, for locations 2 at SNR 10 dB is $\sim$ 7% higher and for location 3 is $\sim$ 4% higher than the FedAvg case. The improvement at locations 2 and 3 is compensated by a slight decrease of $\sim$ 0.5% at location 1. Furthermore, we also see an improvement in Recall. For example, at SNR 20 dB Recall is $\sim$ 4% higher than FL-FedAvg at both UAV locations 2 and 3. As shown in Fig. 11, the improvement in Precision can be seen only at SNR 10 dB. Hence, we base our conclusions on the F1-score, since it is the geometric mean of both Precision and Recall.

Therefore, the numerical results show that both the CL and LL models perform well for all the UAV locations. However, the CL model requires aggregating all data in a central location, which is often impractical. On the other hand, the LL model necessitates maintaining a separate model for each location. While LL models achieve good performance for their specific locations, they lack generalizability to other locations. To address these limitations, we leverage FL as a compromise between CL and LL, where we only share model weights and obtain one final model at the end of the training. We highlight that considering the generalization of the CL model (i.e., testing the CL model performance at a new location from which training data was not collected) is also an interesting research problem that merits further investigation. However, our current results do not address such scenarios, and we defer this investigation to future work.

B. FUSION-BASED SPECTRUM INFERENCE

As shown in Fig. 5, we consider fusing the spectrum hole predictions from multiple UAVs. This is motivated by the fact that individual spectrum sensing performance might fluctuate at different locations, which we observed in the CL, LL, and FL settings. By applying fusion rules, we can compensate for the effect of the different wireless channels observed at different UAV locations. As mentioned earlier, the performance metrics at location 1 are lower in FL compared to locations 2 and 3. Hence, we focus on location 1 to understand the effect of spectrum fusion. To this end, in Table. 2, we compare the F1-score results with and without fusion for location 1 and notice that the overall performance of all configurations is significantly improved by fusion. Furthermore, the proposed pwFedAvg algorithm outperforms FedAvg for all SNRs when we apply fusion. For example, at SNR 10 dB, fusion-based pwFedAvg achieves 95.45%, while fusion-based FedAvg achieves 91.52%. Therefore, fusion-based pwFedAvg further improves the performance by proportionally scaling the weights of models at the training stage as well as fusing the spectrum sensing results from different locations at the testing stage. The comparison results for the spectrum fusion results at locations 2 and 3 are omitted for brevity, as they show similar trends.

C. SPECTRUM RESOURCE ALLOCATION USING RL

As mentioned in Section V, we use deep Q-learning methods for allocating spectrum resources to the UAVs. In Fig. 15a, we compare the training performance of three variants of Q-learning methods for allocating a sub-channel to a single UAV whenever the fusion rule detects at least a single spectrum hole. In this regard, the Q-learning methods choose the best sub-channel to allocate to a UAV. Over time, the RL algorithm identifies which sub-channel to allocate and hence the average aggregated utility for allocating one spectrum hole to a UAV saturates at 200 kb/s. It is observed that DDQN with soft update performs slightly better and converges ear-

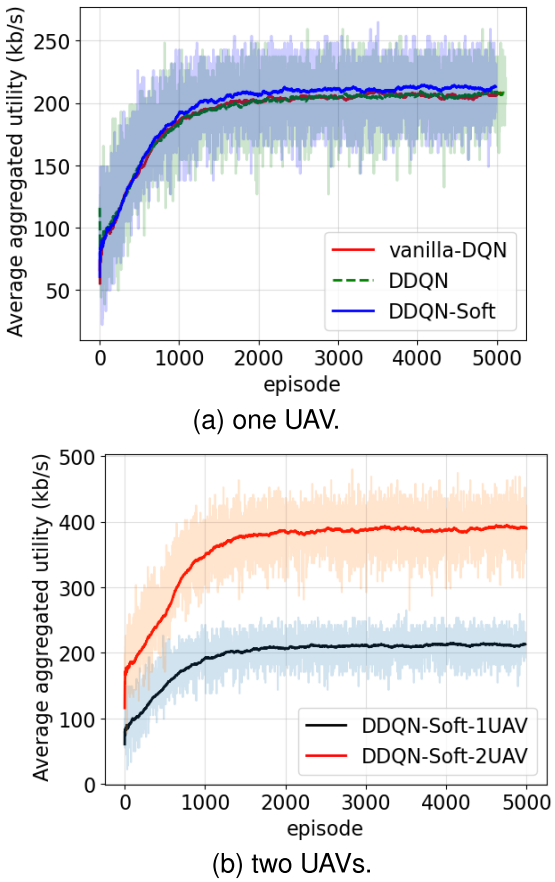


FIGURE 15. **Training results for spectrum resource allocation using RL.In (a), we compare the performance of different Q-learning methods for resource allocation to a single UAV. In (b), we evaluate the performance of DDQN-Soft when allocating resources to one or two UAVs.**

lier than DDQN and vanilla-DQN. While Q-learning-based studies allocate a single resource to a secondary user, we consider allocating spectrum holes to 2 UAVs. In this case, we have augmented the DDQN algorithm with a soft update to generate the two best actions where our action is the best sub-channel or spectrum hole to allocate to the requesting UAVs. In this regard, the DDQN allocates a spectrum hole to two UAVs simultaneously. Hence, from the results in Fig. 15b, we observe that the utility performance with two SUs is slightly less than two times the performance with a single SU. We further note that this paper tries to explore the possibility of integrating spectrum sensing and sharing by making use of the existing RL algorithms. In this paper, since we consider M sub-channels, which correspond to a finite discrete action space, we opted for Q-learning methods. However, other advanced RL algorithms can be explored and integrated into the proposed framework.

VIII. CONCLUSION

In this paper, we developed a collaborative wideband spectrum sensing and sharing solution for networked UAVs. To train machine learning models for detecting spectrum holes, we explored the applications of FL and developed an architecture that integrates wireless dataset generation into the FL model training and aggregation steps. To this end, we proposed the pwFedAvg algorithm to incorporate wireless channel conditions and received signal powers into the FL aggregation algorithm. To further enhance the accuracy of the predicted spectrum holes by individual UAVs, we considered spectrum fusion at the central server. Additionally, by leveraging deep Q-learning methods, the detected spectrum holes are dynamically allocated to the requesting UAVs.

To evaluate the proposed methods, we generated a near-realistic synthetic dataset using MATLAB LTE toolbox by incorporating base-station locations in a chosen area of interest, performing ray-tracing, and emulating the primary users' channel usage in terms of I/Q samples. Based on the collected I/Q datasets, we investigated the performance of three model training algorithms, namely CL, LL, and FL. Furthermore, the proposed pwFedAvg algorithm outperforms FedAvg while achieving comparable results with respect to the CL method without the need for sharing all datasets to a central location. From the fusion results, we noticed that the overall performance improved significantly for all learning configurations, and the implemented DDQN method can provide dynamic spectrum scheduling across requesting UAVs. In future work, we plan to expand the application of our developed solutions to other technologies and spectrum bands (beyond LTE), while incorporating realistic spectrum usage of the incumbent users in those bands (i.e., PUs). Furthermore, we aim to develop an optimization-based approach that jointly considers the received powers from the over-the-air samples, along with other relevant metrics (e.g., communication rounds (E), learning rate γ, batch size, device and data heterogeneity, etc.) to derive optimal aggregation weights.

IX. APPENDIX

Proof of Lemma 1: Since $\nabla L_k(\boldsymbol{\omega}_k^*) = 0$, assumption 1 reduces to $L_k(\boldsymbol{\omega}^*)$ - $L_k(\boldsymbol{\omega}_k^*) \leq \frac{L}{2} ||\boldsymbol{\omega}^* - \boldsymbol{\omega}_k^*||_2^2$. Using the identities of vector norm and max-norm for a vector $\mathbf{x}$, $||\mathbf{x}||_2^2 \leq d||\mathbf{x}||_\infty^2 = d(\max_i |x_i|)^2$, we have:

$$\frac{L}{2}||\boldsymbol{\omega}^* - \boldsymbol{\omega}_k^*||_2^2 \leq \max_k\{\frac{Ld}{2}(\max_i\{|\omega_i^* - \omega_{k,i}^*|\})^2\},$$

which completes the proof.

Proof of Lemma 2: Using the definition of virtual sequences from Eq. (21), we have:

$$\begin{aligned}
&\mathbb{E}\big(||\boldsymbol{a}^t - \bar{\boldsymbol{a}}^t||_2^2\big) \\
&\overset{(a)}{=} \mathbb{E}\Big[\sum_{k=1}^{K} ||\left(\frac{\alpha_k^t}{\alpha^t}\right)\big[\nabla L_k(\boldsymbol{\omega}_k^t; \boldsymbol{\xi}_k^t) - \nabla L_k(\boldsymbol{\omega}_k^t)\big]||_2^2\Big] \\
&\overset{(b)}{\leq} \sum_{k=1}^{K}\left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \mathbb{E}\big[||\nabla L_k(\boldsymbol{\omega}_k^t; \boldsymbol{\xi}_k^t) - \nabla L_k(\boldsymbol{\omega}_k^t)||_2^2\big] \\
&\overset{(c)}{\leq} \sum_{k=1}^{K}\left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \rho_k^2,
\end{aligned} \tag{30}$$

where (a) is from Eq. (21), (b) comes from Jensen's inequality, and (c) is by applying Assumption 3.

Proof of Lemma 3: Similar to [52] in the convergence analysis we define $\boldsymbol{\omega}^{t+1} = \boldsymbol{\omega}^t - \gamma^t \boldsymbol{a}^t$, and using the definition of virtual sequences from Eq. (21) we obtain the following equation:

$$\begin{aligned}||\boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^*||_2^2 &= ||\boldsymbol{\omega}^t - \gamma^t \boldsymbol{a}^t - \boldsymbol{\omega}^*||_2^2 \\ &= ||\boldsymbol{\omega}^t - \gamma^t \bar{\boldsymbol{a}}^t + \gamma^t \bar{\boldsymbol{a}}^t - \gamma^t \boldsymbol{a}^t - \boldsymbol{\omega}^*||_2^2 \\ &= \underbrace{||\boldsymbol{\omega}^t - \gamma^t \bar{\boldsymbol{a}}^t - \boldsymbol{\omega}^*||_2^2}_{A_1} + \underbrace{(\gamma^t)^2 ||\boldsymbol{a}^t - \bar{\boldsymbol{a}}^t||_2^2}_{A_2} \\ &\underbrace{-2\gamma^t \langle \boldsymbol{\omega}^t - \gamma^t \bar{\boldsymbol{a}}^t - \boldsymbol{\omega}^*, \boldsymbol{a}^t - \bar{\boldsymbol{a}}^t \rangle}_{A_3}. \end{aligned} \tag{31}$$

Since $\mathbb{E}[\boldsymbol{a}^t] = \bar{\boldsymbol{a}}^t$, it can be seen that $\mathbb{E}[A3] = 0$. By expanding $\mathbb{E}[A_1]$, we have:

$$\begin{aligned}\mathbb{E}[A_1] = \mathbb{E}\Big[||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2 + \underbrace{(\gamma^t)^2 ||\bar{\boldsymbol{a}}^t||_2^2}_{A_{1,1}} \\ \underbrace{- 2\,\gamma^t \langle \boldsymbol{\omega}^t - \boldsymbol{\omega}^*, \bar{\boldsymbol{a}}^t \rangle}_{A_{1,2}} \Big]\end{aligned} \tag{32}$$

The bound on $A_{1,1}$ term can be derived as follows:

$$\begin{aligned}\mathbb{E}[A_{1,1}] &\overset{(a)}{=} (\gamma^t)^2\, \mathbb{E}\Big[|| \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \nabla L_k(\boldsymbol{\omega}_k^t)||_2^2\Big] \\ &\overset{(b)}{\leq} (\gamma^t)^2 \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2 ||\nabla L_k(\boldsymbol{\omega}_k^t)||_2^2 \\ &\overset{(c)}{\leq} 2L\,(\gamma^t)^2 \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \big(L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}_k^*)\big),\end{aligned} \tag{33}$$

where (a) is from Eq. (21), (b) comes from Jensen's inequality, and (c) is obtained by applying Assumption 1 and L-smoothness property [59]. The bound for $A_{1,2}$ term can be derived as follows:

$$\begin{aligned}&\mathbb{E}[A_{1,2}] \\ &= -2\,\gamma^t\, \langle \boldsymbol{\omega}^t - \boldsymbol{\omega}^*, \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \nabla L_k(\boldsymbol{\omega}_k^t) \rangle \\ &= -2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \langle \boldsymbol{\omega}^t - \boldsymbol{\omega}^*,\ \nabla L_k(\boldsymbol{\omega}_k^t) \rangle \\ &= -2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \langle \boldsymbol{\omega}^t - \boldsymbol{\omega}_k^t,\ \nabla L_k(\boldsymbol{\omega}_k^t) \rangle \\ &\quad - 2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \langle \boldsymbol{\omega}_k^t - \boldsymbol{\omega}^*,\ \nabla L_k(\boldsymbol{\omega}_k^t) \rangle \\ &\overset{(a)}{\leq} \gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \Big[\frac{1}{\gamma^t} ||\boldsymbol{\omega}^t - \boldsymbol{\omega}_k^t||_2^2 + \gamma^t ||\nabla L_k(\boldsymbol{\omega}_k^t)||_2^2\Big] \\ &\quad - 2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \Big[L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}^*) + \frac{\beta}{2} ||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2\Big] \\ &\overset{(b)}{\leq} \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \Big[||\boldsymbol{\omega}^t - \boldsymbol{\omega}_k^t||_2^2 + (\gamma^t)^2 ||\nabla L_k(\boldsymbol{\omega}_k^t)||_2^2\Big] \\ &\quad - 2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \Big[L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}^*) + \frac{\beta}{2} ||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2\Big] \\ &\overset{(c)}{\leq} \underbrace{\sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} ||\boldsymbol{\omega}^t - \boldsymbol{\omega}_k^t||_2^2}_{A_{1,2,1}} + \underbrace{(\gamma^t)^2 \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} ||\nabla L_k(\boldsymbol{\omega}_k^t)||_2^2}_{A_{1,2,2}} \\ &\quad \underbrace{-2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \big[L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}^*)\big]}_{A_{1,2,3}} - \beta\gamma^t\, ||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2\end{aligned} \tag{34}$$

where the first term in (a) results from applying Cauchy-Schwartz inequality and AM-GM inequality [52], [53], and the second term comes from Assumption 2. Furthermore, the second term in (c) comes from the fact that $\sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} = 1$. Using Lemma 3 from [52], $\mathbb{E}[A_{1,2,1}]$ is bounded by $2\gamma_t^2(E-1)^2G^2$. Combining $\mathbb{E}[A_{1,1}]$, $\mathbb{E}[A_{1,2,2}]$ and $\mathbb{E}[A_{1,2,3}]$, we have

$$\begin{aligned}&\mathbb{E}[A_{1,1}] + \mathbb{E}[A_{1,2,2}] + \mathbb{E}[A_{1,2,3}] \\ &\leq 4L(\gamma^t)^2 \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \big(L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}_k^*)\big) \\ &\quad - 2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \big(L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}^*)\big) \\ &\overset{(a)}{\leq} 4L\tau(\gamma^t)^2 \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \\ &\quad - 2\gamma^t \sum_{k=1}^{K} \frac{\alpha_k^t}{\alpha^t} \left[1 - L\gamma^t \frac{\alpha_k^t}{\alpha^t}\right] \big(L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}^*)\big) \\ &\overset{(b)}{\leq} 4\tau\gamma^t \left[1 - 2L\gamma^t \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2\right],\end{aligned} \tag{35}$$

where (a) comes from Lemma 1 and (b) comes from the fact that $L_k(\boldsymbol{\omega}_k^t) - L_k(\boldsymbol{\omega}_k^*) \geq 0$ and Lemma 1. Now, $\mathbb{E}[A_2] = (\gamma^t)^2 \sum_{k=1}^{K} \left(\frac{\alpha_k^t}{\alpha^t}\right)^2 \rho_k^2$ can be found easily by applying Lemma 1. Substituting $\mathbb{E}[A_1]$, $\mathbb{E}[A_2]$, and $\mathbb{E}[A_3]$ into $\mathbb{E}\big(||\boldsymbol{\omega}^{t+1} - \boldsymbol{\omega}^t||_2^2\big)$ and using the fact that $\frac{1}{\kappa} \leq \gamma^t$, we complete the proof.

Proof of Theorem 1 Similar to [49] and [52], we define $\Delta^t = \mathbb{E}[||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2]$. From Lemma 3, it follows that, $\Delta^{t+1} \leq (1 - \beta\gamma^t)\Delta^t + (\gamma^t)^2\zeta^t$. We assume $\gamma^t = \frac{\alpha}{t+\mu}$ for some $\alpha > \frac{1}{\beta}$ and $\mu > 1$. Assuming $\lambda = \max\{\frac{\alpha^2\zeta}{\alpha\beta - 1}, \mu\Delta^0\}$, we will prove $\Delta^t \leq \frac{\lambda}{t+\mu}$ by induction as follows. The definition of λ ensures that the inequality $\Delta^t \leq \frac{\lambda}{t+\mu}$ holds for $t = 0$. For the inequality to hold for $t > 0$, it follows from the definition as follows:

$$\begin{aligned}\Delta^{t+1} &\leq (1 - \beta\gamma^t)\Delta^t + (\gamma^t)^2\zeta^t \\ &\leq \left(1 - \frac{\alpha\beta}{t+\mu}\right) + \frac{\alpha^2\zeta^t}{(t+\mu)^2}\end{aligned}$$

$$\leq \frac{t+\mu-1}{(t+\mu)^2}\lambda + \underbrace{\left[\frac{\alpha^2\zeta^t - \alpha\beta + 1}{(t+\mu)^2}\right]}_{\leq 0}$$
$$\leq \frac{t+\mu-1}{(t+\mu)^2-1}\lambda \leq \frac{\lambda}{(t+1)+\mu}. \quad (36)$$

Specifically, if we choose $\alpha = \frac{2}{\beta}$, $\mu = \frac{2L}{\beta}$, then $\gamma^t = \frac{2}{\beta t + 2L}$. Then, we have

$$\lambda = \max\left\{\frac{\alpha^2\zeta}{\alpha\beta - 1}, \mu\Delta^0\right\}$$
$$\leq \frac{\alpha^2\zeta}{\alpha\beta-1} + \mu\Delta^0 = \frac{4\zeta}{\beta^2} + \frac{2L}{\beta}\,\mathbb{E}||\boldsymbol{\omega}^0 - \boldsymbol{\omega}^*||_2^2. \quad (37)$$

Finally, we have:

$$\mathbb{E}[L(\boldsymbol{\omega}^t)] - L^* \overset{(a)}{\leq} \frac{L}{2}||\boldsymbol{\omega}^t - \boldsymbol{\omega}^*||_2^2 = \frac{L}{2}\Delta^t = \frac{L}{2}\frac{\lambda}{t+\mu}$$
$$\overset{(b)}{\leq} \frac{L}{2\alpha}\frac{\alpha}{t+\mu}\left[\frac{4\zeta}{\beta^2} + \frac{2L}{\beta}\,\mathbb{E}||\boldsymbol{\omega}^0 - \boldsymbol{\omega}^*||_2^2\right]$$
$$\overset{(c)}{\leq} \frac{L}{\beta t + 2L}\left[\frac{2\zeta}{\beta} + L\,\mathbb{E}||\boldsymbol{\omega}^0 - \boldsymbol{\omega}^*||_2^2\right], \quad (38)$$

where *(a)* comes from L-smoothness of the loss function and using the fact that $\nabla L(\boldsymbol{\omega}^*) = 0$, *(b)* is computed using Eq. (37), and *(c)* is computed by substituting the values of α and μ. Hence, the convergence is proved to be $\mathcal{O}(\frac{1}{T})$.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration (NASA) and the National Science Foundation (NSF).

REFERENCES

[1] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, Mar. 2017.

[2] FAA. *Drones by the Numbers*. Accessed: Feb. 11, 2025. [Online]. Available: https://www.faa.gov/node/54496

[3] S. Li et al., "Beyond visual line of sight UAV control for remote monitoring using directional antennas," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[4] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned aircraft system traffic management (UTM) concept of operations," in *Proc. 16th AIAA Aviation Technol., Integr., Oper. Conf.*, Washington, DC, USA, Jun. 2016, doi: 10.2514/6.2016-3292.

[5] A. S. Abdalla and V. Marojevic, "Communications standards for unmanned aircraft systems: The 3GPP perspective and research drivers," *IEEE Commun. Standards Mag.*, vol. 5, no. 1, pp. 70–77, Mar. 2021.

[6] M. Rimjha and A. Trani, "Urban air mobility: Factors affecting vertiport capacity," in *Proc. Integr. Commun. Navigat. Surveill. Conf. (ICNS)*, Apr. 2021, pp. 1–14.

[7] M. Ghazikor, K. Roach, K. Cheung, and M. Hashemi, "Exploring the interplay of interference and queues in unlicensed spectrum bands for UAV networks," in *Proc. 57th Asilomar Conf. Signals, Syst., Comput.*, Oct. 2023, pp. 729–733.

[8] W. S. H. M. W. Ahmad et al., "5G technology: Towards dynamic spectrum sharing using cognitive radio networks," *IEEE Access*, vol. 8, pp. 14460–14488, 2020.

[9] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast wideband spectrum sensing through real-time in-the-loop deep learning," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[10] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[11] Y. Li, W. Zhang, C.-X. Wang, J. Sun, and Y. Liu, "Deep reinforcement learning for dynamic spectrum sensing and aggregation in multi-channel wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 2, pp. 464–475, Jun. 2020.

[12] H. Q. Nguyen, B. T. Nguyen, T. Q. Dong, D. T. Ngo, and T. A. Nguyen, "Deep Q-learning with multiband sensing for dynamic spectrum access," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Oct. 2018, pp. 1–5.

[13] J. Kakar and V. Marojevic, "Waveform and spectrum management for unmanned aerial systems beyond 2025," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[14] B. Shang, V. Marojevic, Y. Yi, A. S. Abdalla, and L. Liu, "Spectrum sharing for UAV communications: Spatial spectrum sensing and open issues," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 104–112, Jun. 2020.

[15] S. R. Chintareddy, K. Roach, K. Cheung, and M. Hashemi, "Collaborative wideband spectrum sensing and scheduling for networked UAVs in UTM systems," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2023, pp. 3064–3069.

[16] B. Shang et al., "Spatial spectrum sensing-based D2D communications in user-centric deployed HetNets," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[17] H. Chen, L. Liu, T. Novlan, J. D. Matyjas, B. L. Ng, and J. Zhang, "Spatial spectrum sensing-based device-to-device cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7299–7313, Nov. 2016.

[18] H. Chen, L. Liu, H. S. Dhillon, and Y. Yi, "QoS-aware D2D cellular networks with spatial spectrum sensing: A stochastic geometry view," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3651–3664, May 2019.

[19] B. Shang, L. Liu, R. M. Rao, V. Marojevic, and J. H. Reed, "3D spectrum sharing for hybrid D2D and UAV networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5375–5389, Sep. 2020.

[20] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, "Deep CM-CNN for spectrum sensing in cognitive radio," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[21] D. Chew and A. B. Cooper, "Spectrum sensing in interference and noise using deep learning," in *Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2020, pp. 1–6.

[22] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[23] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan. 2019.

[24] H. Albinsaid, K. Singh, S. Biswas, and C.-P. Li, "Multi-agent reinforcement learning-based distributed dynamic spectrum access," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 1174–1185, Jun. 2021.

[25] Y. Bokobza, R. Dabora, and K. Cohen, "Deep reinforcement learning for simultaneous sensing and channel access in cognitive networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 7, pp. 4930–4946, Jul. 2023.

[26] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[27] Z. Chen, Y.-Q. Xu, H. Wang, and D. Guo, "Federated learning-based cooperative spectrum sensing in cognitive radio," *IEEE Commun. Lett.*, vol. 26, no. 2, pp. 330–334, Feb. 2022.

[28] Z. Gao, A. Li, Y. Gao, B. Li, Y. Wang, and Y. Chen, "FedSwap: A federated learning based 5G decentralized dynamic spectrum access system," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2021, pp. 1–6.

[29] M. Wasilewska, H. Bogucka, and H. V. Poor, "Secure federated learning for cognitive radio sensing," *IEEE Commun. Mag.*, vol. 61, no. 3, pp. 68–73, Mar. 2023.

[30] N. A. Khalek, D. H. Tashman, and W. Hamouda, "Advances in machine learning-driven cognitive radio for wireless networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1201–1237, 2nd Quart., 2024.

[31] X. Liu, Y. Deng, and T. Mahmoodi, "Wireless distributed learning: A new hybrid split and federated learning approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2650–2665, Apr. 2023.

[32] Z. Wang, Y. Zhou, Y. Shi, and W. Zhuang, "Interference management for over-the-air federated learning in multi-cell wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2361–2377, Aug. 2022.
[33] G. Shi, S. Guo, J. Ye, N. Saeed, and S. Dang, "Multiple parallel federated learning via over-the-air computation," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1252–1264, 2022.
[34] B. Xiao, X. Yu, W. Ni, X. Wang, and H. V. Poor, "Over-the-air federated learning: Status quo, open challenges, and future directions," 2023, *arXiv:2307.00974*.
[35] S. M. Azimi-Abarghouyi and L. Tassiulas, "Over-the-air federated learning via weighted aggregation," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 18240–18253, Dec. 2024.
[36] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of federated learning over a noisy downlink," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1422–1437, Mar. 2022.
[37] X. Wei and C. Shen, "Federated learning over noisy channels: Convergence analysis and design examples," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 1253–1268, Jun. 2022.
[38] Y. Mu, C. Shen, and Y. C. Eldar, "Optimizing federated averaging over fading channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 1277–1281.
[39] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6804–6819, Oct. 2021.
[40] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, 2020.
[41] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE ICASSP*, May 2020, pp. 8851–8855.
[42] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2011, pp. 205–216.
[43] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[44] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
[45] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 146–152, Sep. 2018.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[48] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, A. Singh and J. Zhu. Eds., Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.
[49] N. Yan, K. Wang, C. Pan, and K. K. Chai, "Performance analysis for channel-weighted federated learning in OMA wireless networks," *IEEE Signal Process. Lett.*, vol. 29, pp. 772–776, 2022.
[50] JMLB. (2017). *Calculate the Gradient of the Softmax Function*. Accessed: Oct. 2, 2024. [Online]. Available: https://jmlb.github.io/ml/2017/12/26/Calculate_Gradient_Softmax/
[51] P. Unleashed. (2022). *Derivation of the Binary Cross-Entropy Loss Gradient*. [Online]. Available: https://www.python-unleashed.com/post/derivation-of-the-binary-cross-entropy-loss-gradient
[52] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, *arXiv:1907.02189*.
[53] M. Salehi and E. Hossain, "Federated learning in unreliable and resource-constrained cellular wireless networks," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 5136–5151, Aug. 2021.
[54] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2016, vol. 30, no. 1, pp. 2094–2100.
[55] *CellMapper*. Accessed: Feb. 11, 2025. [Online]. Available: https://www.cellmapper.net/map
[56] *OpenStreetMap*. Accessed: Feb. 11, 2025. [Online]. Available: https://www.openstreetmap.org
[57] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," 2020, *arXiv:2008.05756*.
[58] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online deep learning: Learning deep neural networks on the fly," 2017, *arXiv:1711.03705*.
[59] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

SRAVAN REDDY CHINTAREDDY (Graduate Student Member, IEEE) received the M.Sc. degree from NYU Tandon School of Engineering in 2020. He is a Ph.D. student with the University of Kansas (KU). His current research interests include millimeter-wave (mmWave) and THz communications, network planning and optimization, software defined radios, and spectrum sensing and sharing for unmanned aerial vehicles (UAVs). He is a member of the Institute for Information Sciences (I2S) at KU.

KEENAN ROACH received the B.S. and M.S. degrees in mechanical engineering from Kansas State University in 2001 and 2002, respectively. He worked on the control law team for the F-35 Joint Strike Fighter jet before focusing on air traffic management research. For the past 19 years he has served at NASA's North Texas Research Station working with the FAA and industry partners to improve air traffic management in the National Airspace System.

KENNY CHEUNG (Member, IEEE) received the B.S. degrees in both aeronautical engineering and mechanical engineering from University of California, Davis, in 1993, and the M.S. degree in aeronautics and astronautics from Stanford University, in 1995. He was the Technical Area Lead and a Principal Engineer of the USRA Aero flight dynamics technical area under the NASA Academic Mission Services (NAMS) contract and is currently a Senior Research Associate under San Jose State University Research Foundation at NASA Ames Research Center. He serves as the Chief Software Architect for the flight control design tools developed under the U.S. Army Technology Development Directorate at Moffett Field, California. He has over 20 years of hands-on Aerospace software application development experience, in the area of flight control design and optimization for rotorcraft, fixed-wing aircraft, and UAS/drones. He has authored and co-authored over 30 technical publications on the application of the flight control tools, including the technical book titled Practical Methods for Aircraft and Rotorcraft Flight Control Design: An Optimization Based Approach under the AIAA Education Series (AIAA, 2017).

MORTEZA HASHEMI (Senior Member, IEEE) received the B.S. degree from Sharif University of Technology in 2011, and the M.Sc. and Ph.D. degrees in electrical engineering from Boston University, in 2013 and 2015, respectively. He is an Assistant Professor with the Department of Electrical Engineering and Computer Science, University of Kansas (KU), Lawrence, Kansas. Before joining KU in 2019, he was a Postdoctoral Researcher and a Senior Lecturer at Ohio State University. His research interests span the areas of wireless communications, information systems, real-time data networking, and networked cyber-physical systems. In 2021 and 2022, he was the recipient of the Bellows Scholar Award from the University of Kansas. In 2014, he received the Boston University Provost's Award for excellence in research.