Dynamic Client Scheduling Enhanced Federated Learning for UAVs

Yubo Peng, Feibo Jiang, *Senior Member, IEEE*, Siwei Tu, Li Dong, Kezhi Wang, *Senior Member, IEEE*, and Kun Yang, *Fellow, IEEE*.

***Abstract*—Although Federated Learning (FL) applied in Unmanned Aerial Vehicles (UAVs) offers substantial benefits, it also poses some challenges. These challenges arise primarily from the dynamic nature of UAV movements and the constraints imposed by limited wireless channel resources. This leads to the situation where only partial UAVs can participate in the FL process during each communication round, introducing the bias of the optimization objective that adversely impacts model accuracy. To address this issue, we introduce a Multi-action Q Network (MQN) for client scheduling, which selects suitable UAVs for each round, resolving the problems of the partial participation of UAVs. Furthermore, we propose a Gain-based Parameter Aggregation (GPA), which assigns a "contribution score" to each local model based on its contribution, correcting the bias of the optimization objective in FL. Simulation results demonstrate the effectiveness of the proposed methods.**

***Index Terms*—Federated learning; deep reinforcement learning; wireless communications; unmanned aerial vehicles**

I. Introduction

UNMANNED Aerial Vehicles (UAVs) have attracted substantial interest across various fields such as healthcare, logistics, agriculture, and transportation. These fields leverage UAVs equipped with Deep Learning (DL) capabilities to perform a variety of tasks [1], [2], including trajectory planning, resource allocation, intrusion detection, and aerial target identification [3]. However, due to the high altitude and mobility of UAVs, maintaining consistent connections between the UAVs and ground base stations is not always feasible. This poses a significant challenge for executing learning-related tasks using traditional centralized DL approaches. Particularly when transmitting large volumes of data over aerial links, these challenges can lead to potential data privacy breaches and task latency issues.

This work was supported in part by the National Natural Science Foundation of China under Grant 41904127 and 62132004, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2024JJ5270, in part by the Open Project of Xiangjiang Laboratory under Grant 22XJ03011, and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 22B0663. (Corresponding authors: Feibo Jiang.)

Yubo Peng and Siwei Tu (pengyubo@hunnu.edu.cn, tusw0411@outlook.com) are with School of Information Science and Engineering, Hunan Normal University, Changsha, China.

Feibo Jiang (jiangfb@hunnu.edu.cn) is with Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, China.

Li Dong (Dlj2017@hunnu.edu.cn) is with Changsha Social Laboratory of Artificial Intelligence, Hunan University of Technology and Business, Changsha, China.

Kezhi Wang (Kezhi.Wang@brunel.ac.uk) is with the Department of Computer Science, Brunel University London, UK.

Kun Yang (kyang@ieee.org) is with the School of Intelligent Software and Engineering, Nanjing University, Suzhou, China.

As a distributed and privacy-preserving machine learning method, Federated Learning (FL) [4] offers a solution to the above challenges. The common Federated Averaging (FedAvg) algorithm [4] enables each client to gather local data and independently train the DL model by computing the gradient on their device. Subsequently, each client only uploads the local model to the parameter server for aggregation. Following this, the parameter server downloads the updated global model to all clients for local model updates. However, when applying FL in the UAVs, there are two critical challenges to be addressed:

1) The limited wireless channel resources (e.g., spectrum and bandwidth) restrict the number of UAVs participating in FL. Moreover, the dynamic nature of UAV flight trajectories results in a changing topology with each communication round. This leads to the unstable consumption of communication energy between UAVs.
2) The common FedAvg weighs the importance of each local model based on the size of the local dataset. However, when only a part of the UAVs can participate in FL and the participated UAVs are uncertain, their weights change in each round. This might lead to the bias of the gradient and the optimization objective [5].

To tackle these challenges and enhance FL in UAVs, we propose the following contributions:

1) We introduce a Multi-action Q Network (MQN) for client scheduling dynamically, in which a minimal number of UAVs that meet the constraints of available spectrum resources are selected in each communication round, according to the current network topology of UAVs and the learning performance of each DL model deployed on UAVs. Thus, we can dynamically adjust the UAVs participating in FL.
2) We present a Gain-based Parameter Aggregation (GPA), to reduce the bias of the optimization objective caused by partial participation of UAVs in FL. This approach assigns a "contribution score" to each participating UAV based on its contribution toward enhancing the global model, thus improving the accuracy of the global model.

II. System Model

As shown in Fig. 1, we consider a multi-UAV cooperative aerial FL system in a resource-limited wireless environment. There is a set of UAVs $\mathcal{K} = \{1, 2, ..., K\}$ in the system, and each UAV equips a single antenna. We assume that the available bandwidth resource is limited and can not support

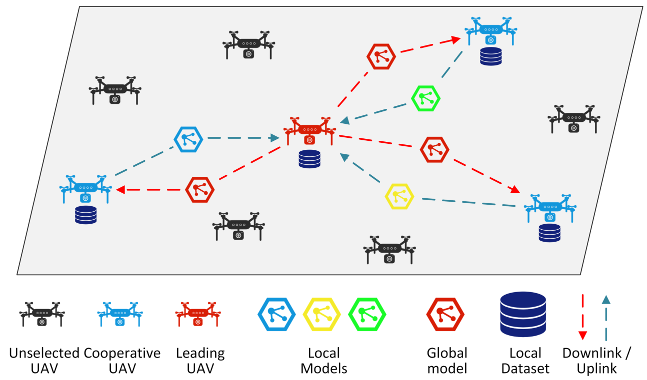


Fig. 1: A multi-UAV cooperative aerial FL system.

that all the UAVs participate in FL in each communication round. Therefore, we have to select part of UAVs to perform FL in each communication round. We divide the overall available bandwidth resources based on Orthogonal Frequency Division Multiplexing (OFDM) and evenly allocate these resources to the selected UAVs for communications. We divide the selected UAVs into two types: cooperative and leading UAVs. The cooperative UAVs utilize their local dataset to train the local models. The leading UAV aggregates local models uploaded from cooperative UAVs and generates a global model. Thereafter, the global model is sent back to cooperative UAVs to update their local models.

A. FL Model

The local dataset of the k-th UAV is denoted as $\mathcal{D}_k = \{(x_{k,1}, y_{k,1}), \ (x_{k,2}, y_{k,2}), ..., \ (x_{k,N_k}, y_{k,N_k})\}$, where the N_k is the number of collected samples in $\mathcal{D}_k$. $x_{k,n}$ and $y_{k,n}$ are the input and output of the n-th samples in $\mathcal{D}_k$, respectively.

For the k-th UAV on its local dataset $\mathcal{D}_k$, the training loss in the t-th step of the training phase can be calculated as:

$$F_k\left(\mathbf{w}_{k,t}\right) = \frac{1}{N_k}\sum_{i=1}^{N_k} f\left(\mathbf{w}_{k,t}, x_{k,i}, y_{k,i}\right) \tag{1}$$

where $f(\mathbf{w}_{k,t}, x_{k,i}, y_{k,i})$ is the training loss function for the i-th sample $(x_{k,i}, y_{k,i})$ in $\mathcal{D}_k$; $\mathbf{w}_{k,t}$ is the weights of the k-th local model in the t-th communication round.

In traditional FL, to ensure the privacy of local data, the global model on the leading UAV is updated by aggregating local models from the cooperative UAVs. This process can be expressed as follows:

$$\mathbf{w}_{g,t} = \frac{1}{N}\sum_{k=1}^{K} N_k \mathbf{w}_{k,t} \tag{2}$$

where $\mathbf{w}_{g,t}$ is the global model on the leading UAV in the t-th communication round; $N = \sum_{k=1}^{K} N_k$ represents the sum data size of all clients.

FL aims to get an optimal global model $\mathbf{w}_{g,t}$ that can minimize the local FL loss of all clients, thus achieving global optimization. Hence, the global loss function of FL can be given by:

$$F_g\left(\mathbf{w}_{g,t}\right) = \frac{1}{K}\sum_{k=1}^{K} F_k\left(\mathbf{w}_{g,t}\right) \tag{3}$$

where $F_k\left(\mathbf{w}_{g,t}\right)$ represents that the local FL loss of the k-th UAV based on $\mathbf{w}_{g,t}$.

B. Transmission Model

For the UAV-to-UAV data transmission in the sky, we adopt the free space path loss model, which is dominated by Line-of-Sight (LoS) conditions. Therefore, when the k-th UAV performs communication with the leading UAV in the t-th communication round, the channel gain $H_{k,t}$ can be calculated by:

$$H_{k,t} = (d_{\mathrm{L},k}(t))^{-\mu} \tag{4}$$

where μ denotes the path loss exponent. $d_{\mathrm{L},k}(t) = \sqrt{(p^{\mathrm{x}}_{\mathrm{L},t} - p^{\mathrm{x}}_{k,t})^2 + (p^{\mathrm{y}}_{\mathrm{L},t} - p^{\mathrm{y}}_{k,t})^2}$ represents the distance between the leading UAV and the k-th UAV in the t-th communication round, where $(p^{\mathrm{x}}_{\mathrm{L},t}, p^{\mathrm{y}}_{\mathrm{L},t})$ represents the positions of the leading UAV and $(p^{\mathrm{x}}_{k,t}, p^{\mathrm{y}}_{k,t})$ represents the positions of the k-th UAV. Since the $H_{k,t}$ is distance-dependent, it is varied in each communication round.

When the k-th UAV uploads its local model to the leading UAV, the uplink rate can be given by [6]:

$$v^{\mathrm{u}}_k = B_{\mathrm{L},k}\log_2\left(1 + \frac{H_{k,t}P_k}{\sigma^2}\right) \tag{5}$$

where P_k represents the transmit power of the k-th UAV. σ^2 is the noise power of the additive white Gaussian noise. $B_{\mathrm{L},k}$ is the allocated bandwidth between the leading UAV and the k-th UAV.

Similarly, when the leading UAV transmits the global model to the k-th UAV, the downlink data rate can be defined as:

$$v^{\mathrm{d}}_k = B_{\mathrm{L},k}\log_2\left(1 + \frac{H_{k,t}P_{\mathrm{d}}}{\sigma^2}\right) \tag{6}$$

where P_{d} is the transmitting power of the leading UAV.

Given the uplink data rate v^{u}_k and the downlink data rate v^{d}_k, the transmission delays between the k-th UAV and the leading UAV over uplink and downlink are respectively specified as follows:

$$\tau^{\mathrm{u}}_k = \frac{Z\left(\mathbf{w}_{k,t}\right)}{v^{\mathrm{u}}_k}; \ \tau^{\mathrm{d}}_k = \frac{Z\left(\mathbf{w}_{g,t}\right)}{v^{\mathrm{d}}_k} \tag{7}$$

where function $Z\left(\mathbf{w}_{k,t}\right)$ represents the number of the bits that the k-th UAV requires to transmit its local model $\mathbf{w}_{k,t}$ to the leading UAV, while $Z\left(\mathbf{w}_{g,t}\right)$ is the number of the bits that the leading UAV requires transmitting the global model $\mathbf{w}_{g,t}$ to the k-th UAV.

C. Energy Model

In this letter, we mainly consider the energy consumption of transmission when uploading and downloading models. We assume that the leading UAV transmits the global model to cooperative UAVs based on a multicast way. Therefore, for the k-th UAV, the energy consumption of transmission in the t-th communication round can be calculated by:

$$E_{k,t} = \begin{cases} \tau_k^{\mathrm{d}} \cdot P_d + \rho, & \text{if UAV } k \text{ is the leading UAV,} \\ \tau_k^{\mathrm{u}} \cdot P_k + \rho, & \text{if UAV } k \text{ is the cooperative UAV,} \\ \rho, & \text{if UAV } k \text{ is the idle UAV,} \end{cases} \tag{8}$$

where ρ denotes the energy consumption of UAV hovering when performing model transmission.

D. Problem Formulation

In this letter, we aim to select appropriate UAVs for participation in FL in each communication round. We use a binary variable, $C_{k,t}$, to denote the state of the k-th UAV during the t-th communication round. $C_{k,t} = 1$ represents that the k-th UAV can participate in FL during the t-th communication round, and $C_{k,t} = 0$ otherwise. We apply $\mathcal{C}_t = \{C_{k,t} | k \in \mathcal{K}\}$ to denote the state set of all UAVs in the t-th communication round. Considering the limitation of bandwidth resources, in each communication round, we assume that the number of UAVs participating in FL is M. Our goals are to minimize the global loss function of FL and the energy consumption during the entire FL training process. Therefore, the objective function can be defined as follows:

$$\min_{\mathcal{C}_t, \mathbf{w}_{g,T}} F_g\left(\mathbf{w}_{g,T}\right) + \zeta \sum_{t=1}^{T} \sum_{k=1}^{K} E_{k,t} \tag{9a}$$

$$\text{s.t. } C_{k,t} \in \{0,1\}, \ \forall k \in \mathcal{K} \tag{9b}$$

$$\sum_{k=1}^{K} C_{k,t} = M \tag{9c}$$

where T represents the total communication rounds in the FL; $\mathbf{w}_{g,T}$ represents the global FL model parameter in the T-th communication round, $F_g\left(\mathbf{w}_{g,T}\right)$ is the final global FL loss based on $\mathbf{w}_{g,T}$; ζ is a coefficient that adjusts the sensitivity of $E_{k,t}$ so that $F_g\left(\mathbf{w}_{g,T}\right)$ and $E_{k,t}$ are relatively balanced; Eq. (9b) and Eq. (9c) indicate that only a part of UAVs can participate in FL in each communication round.

III. MQN and GPA Enhanced FL in UAVs

In this section, we introduce MQN to select suitable UAVs for performing FL. In addition, we propose GPA to provide a more effective parameter aggregation when only partially uncertain UAVs participate in FL.

A. MQN

As illustrated in Fig. 2, MQN improves the network structure of dueling DQN [7], which can more accurately evaluate the benefits brought by all participating UAVs. However, different from the traditional dueling DQN which only takes one of the most valuable actions as the output, in MQN, the top-M actions with the highest predicted values are all taken as outputs. Moreover, to prevent overestimating the values of the selected top-M actions, MQN introduces double Q networks to estimate the Q values [8]. The implementation process of MQN is divided into the following steps:

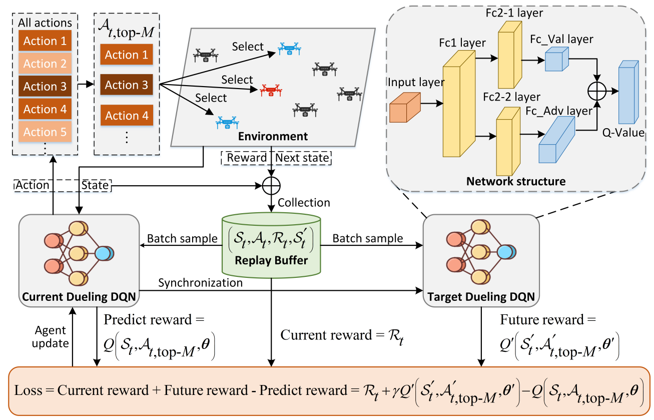


Fig. 2: Flowchart of MQN.

1) State Acquisition: We define the state as:

$$\mathcal{S}_t = \{\mathbf{F}_t, \mathbf{P}_t\} \tag{10}$$

where $\mathbf{F}_t = \left[F_k\left(\mathbf{w}_{k,t}\right) | \ k \in \mathcal{K}\right]$ denotes the local FL loss for each UAV, which represents the learning performance of the individual local models. $\mathbf{P}_t = [(p_{k,t}^{\mathrm{x}}, p_{k,t}^{\mathrm{y}}) | k \in \mathcal{K}]$ signifies the positions of all UAVs in the system, which captures information about the network topology of the UAVs.

2) Action Selection: $\mathcal{S}_t$ is fed into the current dueling Q network and output the values of all actions. We use $\mathcal{A}_t = \{a_{k,t} | k \in \mathcal{K}\}$ to denote the set of all actions, where $a_{k,t}$ represents an action that select the k-th UAV to participate in FL in the t-th round, namely $C_{k,t} = 1$. Due to applying the structure of Dueling DQN, the value of adopting $a_{k,t}$ can be given by:

$$Q(\mathcal{S}_t, a_{k,t}; \boldsymbol{\theta}) = V(\mathcal{S}_t; \boldsymbol{\theta}_\beta) + \left(A(\mathcal{S}_t, a_{k,t}; \boldsymbol{\theta}_\alpha) - \frac{1}{K} \sum_{j \in K} A\left(\mathcal{S}_t, a_{j,t}; \boldsymbol{\theta}_\alpha\right) \right) \tag{11}$$

where $\boldsymbol{\theta}$ is the vector of the Q network parameters; $\boldsymbol{\theta}_\alpha$ is the vector of the parameters of the "advantage Fully Connected (FC) layers" in the Q network; $\boldsymbol{\theta}_\beta$ is the vector of the parameters of the "value FC layers" [7] in the Q network; $A(\cdot)$ is the output of the "advantage FC layers"; $V(\cdot)$ is the output of the "value FC layers".

Then, we select the top-M actions with the highest predicted values as outputs, and the numbered UAVs corresponding to their index values are selected to participate in the FL training. The set of top-M actions can be given by:

$$\mathcal{A}_{t,\text{top-}M} = \arg\max_{\mathcal{A}_{t,\text{top-}M} \subset \mathcal{A}_t} \left(\sum_{a_{k,t} \in \mathcal{A}_{t,\text{top-}M}} Q\left(\mathcal{S}_t, a_{k,t}, \boldsymbol{\theta}\right) \right). \tag{12}$$

Based on the top-M actions, we can obtain the set of selected UAVs, denoted by $\mathcal{U}_t$. $\mathcal{U}_t = \{k | a_{k,t} \in \mathcal{A}_{t,\text{top-}M}\}$. Then, we use the K-medoids algorithm [9] to select the leading UAV, which satisfies:

$$\sum_{k \in \mathcal{U}_t} d_{\mathrm{L},k}(t) = \min \left(\{ \sum_{j \in \mathcal{U}_t, j \neq i} d_{i,j}(t) | i \in \mathcal{U}_t \} \right) \tag{13}$$

where $d_{i,j}(t)=\sqrt{(p^{\mathrm{x}}_{i,t}-p^{\mathrm{x}}_{j,t})^2+(p^{\mathrm{y}}_{i,t}-p^{\mathrm{y}}_{j,t})^2}$ represents the distance between the i-th UAV and the j-th UAV. Obviously, the UAV with the smallest sum of distances to other selected UAVs is the leading UAV.

3) Reward Feedback: After cooperative UAVs finish their local FL learning, a reward $\mathcal{R}_t$ can be obtained by:

$$\mathcal{R}_t=-\sum_{k\in\mathcal{U}_t}\left(F_k(\mathbf{w}_{k,t})+\iota_1 E_{k,t}+\iota_2\Phi_{k,t}\right) \tag{14}$$

where ι_1 and ι_2 are adjustment coefficients; $\Phi_{k,t}=\|\mathbf{w}_{g,t-1}-\mathbf{w}_{k,t}\|^2$ is used to capture the distance between the local and global models, which ensures the local models do not deviate too much from the global model due to the data heterogeneity [10].

To mitigate the problem of the over-optimistic value estimates, referring to the double DQN algorithm [8], we use the following target:

$$y_t=\mathcal{R}_t+\frac{1}{M}\sum_{a'_{k,t}\in\mathcal{A}'_{t,\text{top-}M}}\gamma Q'\left(\mathcal{S}'_t,a'_{k,t},\boldsymbol{\theta}'\right) \tag{15}$$

where $\boldsymbol{\theta}'$ is the parameter weights of the target network Q'; $\mathcal{A}'_{t,\text{top-}M}$ is the top-M actions obtained under the condition of the next state $\mathcal{S}'_t$; γ is the discount factor for future reward. We select the top-M actions with the highest Q values from the current network Q and then substitute it into the target network Q' to calculate the corresponding value.

4) Agent Update: After local training, we obtain a group of $(\mathcal{S}_t,\mathcal{A}_t,\mathcal{R}_t,\mathcal{S}'_t)$ and then we store it in the replay buffer as a transition. When the amount of transitions in the replay buffer reaches the minimum batch size required for training, the Q network will start training and update its parameter $\boldsymbol{\theta}$. Transitions will be added to the replay buffer until it is full, and then it will overwrite the old transition in this buffer by the First In First Out (FIFO) policy. In addition, the parameter weights of the target network Q' will be synchronized with the current network Q according to a certain frequency.

B. GPA

The widely used FedAvg usually weighs the importance of each local model based on the size of the training data on the corresponding client. However, in this scenario, only a part of the UAVs can participate in each communication round and the participated UAVs are uncertain. Hence, their weights change in each communication round, which might lead to the bias of gradient and optimization objective [5]. As a result, the GPA is proposed, which can assign a "contribution score" to each selected UAV according to its contribution to improving the global model.

We define the "contribution score" $S_{k,t}$ to capture the performance gain of the k-th local model to the global model in the t-th communication round, which can be formulated as:

$$S_{k,t}=\frac{F_k\left(\mathbf{w}_{g,t-1}\right)}{F_k\left(\mathbf{w}_{k,t}\right)} \tag{16}$$

where $F_k\left(\mathbf{w}_{g,t-1}\right)$ represents that the local FL loss based on the $(t-1)$-th global model $\mathbf{w}_{g,t-1}$. The larger $S_{k,t}$ is, the greater the performance gain of the local model over the global model. Hence, the improved parameter aggregation can be given by:

$$\mathbf{w}_{g,t}=\frac{\sum\limits_{k\in\mathcal{U}_t}\mathbf{w}_{k,t}S_{k,t}N_k}{\sum\limits_{k\in\mathcal{U}_t}S_{k,t}N_k}. \tag{17}$$

In this approach, we use the "contribution score" to quantify the gain of the local models to the global model and refer it to weight the parameters of local models. Thus, the bias of the optimization objective caused by the partial participation of UAVs in the FL training is reduced and the learning performance of the global model is enhanced.

IV. Simulations

In this section, some simulations are carried out to evaluate the effectiveness of the proposed methods.

A. Simulation Settings

We assume each local model performs a classification task-based local training, hence MNIST [11] and CIFAR10 [12] are applied to evaluate the proposed methods.

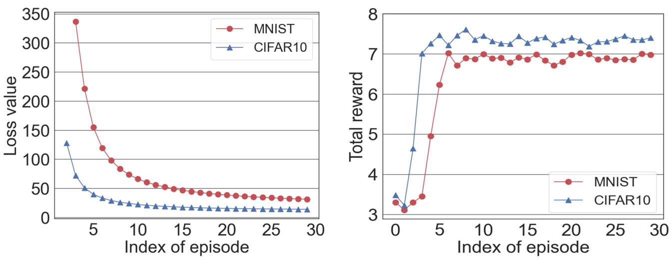
In the system model, the total number of UAVs is set as $K=20$. These UAVs move randomly at an approximately constant speed in an area of 1 km$\times$1 km in each communication round. We randomly divide the MNIST and CIFAR10 datasets into 20 parts and then assign them to the 20 UAVs as their local data. The number of participated UAVs in each communication round is set as $M=5$. The noise power σ^2 is set to -90 dBm. The path loss exponent μ is set to 2. The energy consumption of UAV hovering ρ is set to 0 J. The transmission power and bandwidth of each selected UAV are set as $P_k=P_{\mathrm{d}}=1$ W and $B_{\mathrm{L},k}=1$ MHz, respectively.

For the network structure of MQN, we use one FC layer as the backbone that has 10,368 neurons. We use one FC layer as the "advantage FC layer" that contains 2,580 neurons and one FC layer as the "value FC layer" that has 128 neurons. All FC layers are followed by a Rectified Linear Unit (ReLU). In the training settings of MQN, we set the learning rate to 0.005, the batch size to 64, and the size of the replay buffer to 600. The discount factor is set as $\gamma=0.9$. In the training settings of FL, we set the total number of communication rounds to 60, the learning rate to 0.001, and the batch size to 100.

B. Performance Evaluation

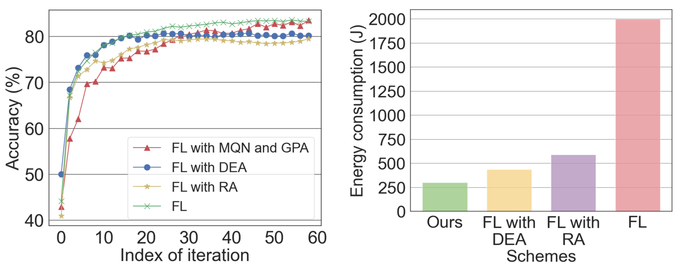
We compare the performance of our proposed methods with several baseline algorithms which include FL with Differential Evolution Algorithm (DEA), FL with Random Selection (RA), and standalone FL.

In Fig. 3, we show the training results of MQN, showcasing the dynamic evolution of loss and total reward across iterations. Notably, MQN achieves convergence after approximately 10 episodes, signifying successful model training. Additionally, Fig. 3(b) highlights a substantial increase in rewards, indicating a continuous enhancement in decision benefits provided by MQN.



(a) (b)

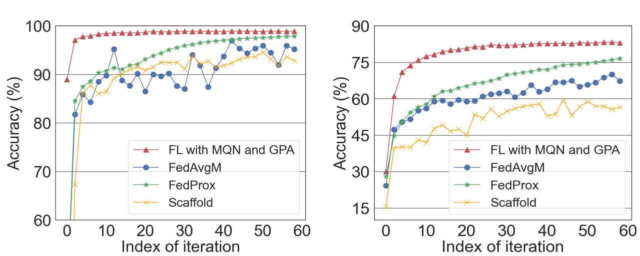
Fig. 3: Training results of MQN on two datasets. (a) Loss results. (b) Reward results.



(a) (b)

Fig. 4: Performance evaluation of different schemes. (a) Accuracy results. (b) Energy consumption results.

Fig. 4 compares the performances of FL with MQN and GPA (ours) against FL with DEA, FL with RA, and standalone FL in terms of accuracy and total energy consumption. Here, the accuracy in Fig. 4 denotes the mean classification accuracy of the local models in the selected UAVs, while energy consumption represents the total energy utilized for communication throughout the training process. It is important to note that our method, FL with DEA, and FL with RA selectively involves a subset of UAVs for each round of FL training, whereas standalone FL encompasses all UAVs. From Fig. 4(a), it is obvious that models trained by standalone FL and ours display superior accuracy. Conversely, models trained via either FL with RA or FL with DEA exhibit lower accuracy. Furthermore, Fig. 4(b) shows that standalone FL consumes the highest communication energy during training, whereas ours consumes the least.



(a) (b)

Fig. 5: Accuracy results with different iterations on two datasets. (a) MNIST. (b) CIFAR10.

Finally, to evaluate the efficacy of our proposed method under conditions where local data is non-independently and identically distributed (non-IID), we compare it with several model aggregation algorithms tailored for such conditions. The compared methods include FedAvgM [13], FedProx [10], and Scaffold [14]. Fig. 5 illustrates the comparative results, in which our method surpasses the others on both datasets, demonstrating the capability of MQN and GPA to tackle non-IID challenges within dynamic UAV environments.

V. Conclusion

In this letter, we introduce MQN and GPA to enhance FL in UAVs. MQN strategically selects suitable UAVs for each FL round, thereby optimizing learning efficiency while meeting spectrum resource constraints. Moreover, the GPA is proposed to assign a gain-based weight to each participating UAV. This weight corresponds to the contribution of the UAV towards the global model during parameter aggregation, thus improving the model accuracy. Simulation results validate the effectiveness of these proposed methods.

References

[1] F. Jiang, L. Dong, Y. Peng, K. Wang, K. Yang, C. Pan, D. Niyato, and O. A. Dobre, "Large language model enhanced multi-agent systems for 6G communications," *arXiv preprint arXiv:2312.07850*, 2023.

[2] F. Jiang, Y. Peng, L. Dong, K. Wang, K. Yang, C. Pan, and X. You, "Large AI model-based semantic communications," *IEEE Wireless Communications*, pp. 1–8, 2024.

[3] F. Syed, S. K. Gupta, S. Hamood Alsamhi, M. Rashid, and X. Liu, "A survey on recent optimal techniques for securing unmanned aerial vehicles applications," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, p. e4133, 2021.

[4] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

[5] X. Yao, T. Huang, R.-X. Zhang, R. Li, and L. Sun, "Federated learning with unbiased gradient aggregation and controllable meta updating," *arXiv preprint arXiv:1910.08234*, 2019.

[6] B. Yang, H. Shi, and X. Xia, "Federated imitation learning for UAV swarm coordination in urban traffic monitoring," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 6037–6046, 2023.

[7] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[8] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[9] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[11] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," *Advances in neural information processing systems*, vol. 2, 1989.

[12] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, "Cifar10-dvs: an event-stream dataset for object classification," *Frontiers in neuroscience*, vol. 11, p. 244131, 2017.

[13] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[14] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.