# Energy Efficient User Scheduling for Hybrid Split and Federated Learning in Wireless UAV Networks

Xiaolan Liu*, Yansha Deng †, Toktam Mahmoodi †
* Loughborough University, †King's College London
Email:*xiaolan.liu@lboro.ac.uk, †{yansha.deng, toktam.mahmoodi}@kcl.ac.uk

***Abstract*—The use of unmanned aerial vehicles (UAVs) as flying users provides various applications by exploiting machine learning (ML) algorithms. Recently, distributed learning algorithms, federated learning (FL) and split learning (SL), have been exploited to train ML models distributedly via sharing model parameters rather than large raw datasets in the conventional centralized learning algorithms. Considering the diversity of users with heterogeneous resources, computation capabilities, and data distributions, we propose a hybrid split and federated learning (HSFL) framework that allows users to select either split training (ST) or federated training (FT) method based on the characteristics of the users in wireless UAV networks. Due to unreliable wireless channels and the limited energy of the users, we further formulate a user scheduling and training method selection problem within HSFL framework as a Multiple-Choice Knapsack Problem (MCKP) and propose an energy-efficient user scheduling algorithm to select a subset of users in each round and schedule each user with either ST or FT method. The simulations demonstrate that our proposed HSFL framework consumes less energy while having the same good test accuracy performance compared to the currently distributed learning algorithms, and the proposed user scheduling algorithm achieves energy-efficient selection of ST or FT method under different distributions.**

## I. Introduction

Wireless unmanned aerial vehicles (UAVs) network is foreseen to be an integral component in the upcoming sixth-generation (6G) networks [1], [2], which has the potential to support various applications, such as video streaming and disaster surveillance. In these scenarios, the UAVs (users) fly over the target area under the control of the base stations (BSs) to collect data (e.g., images and videos) and then transmit them to the BS for data processing. Each user can observe its local environment and collects a sub-dataset that only contains partial environment information, so all the users should transmit their local sub-datasets to the BS for integrated data processing (e.g., training machine learning (ML) models, including deep neural network (DNN) and convolutional neural network (CNN)). However, the transmission of large datasets causes high communication overhead and large energy consumption for the users, and may potentially reveal user privacy [3]. Thanks to the increased computational capability brought by GPUs, distributed learning becomes more attractive by enabling local learning at users and model aggregation at the BS via only sharing model parameters rather than raw data.

Federated learning (FL) was first investigated as a distributed learning approach in wireless networks, which allows the users to perform local learning and only send the local model updates instead of the whole dataset to the BS for model aggregation [4]. FL adopts a parallel model training mechanism, where all the users receive the global ML model from the BS and perform local training with their local sub-datasets simultaneously, and then send their local model updates to the BS for performing model aggregation. In FL, all the users are required to have powerful computational capabilities, and the communication overhead of which depends on the model size. Different from FL, split learning (SL) divides the ML model, e.g., DNN into several sub-models by the cut layer and distributes them to different entities for model training (e.g., a sub-model at user, namely UE-side model, a sub-model at BS, called BS-side model), wherein only the smashed data of the cut layer is shared [5]. To this end, SL limits the UE-side model down to a few layers, thus reducing the computational overhead of the user compared to FL [6]. The communication overhead of SL depends on the size of the dataset owned by the user. In practice, only deploying FL or SL may not be efficient due to the diversity of the users with heterogeneous resources, various computational capabilities, and data distributions. In [7], split federated learning (SFL) was proposed to combine the advantages of both FL and SL, which still faces the problem of large communication overhead as in SL because all the users adopt the SL method.

In either SL or FL, the users must transmit their learning parameters over wireless links, in which the learning performance can be affected due to limited wireless resources (e.g., time, bandwidth, and energy). Specifically, the limited bandwidth restricts the number of users sending their learning parameters in each round, which requires the design of a user scheduling scheme. Moreover, the limited energy of the user brings new challenges for deploying distributed learning algorithm, the authors in [8] formulated an energy minimization problem that jointly considers both communication and computation optimization, which effectively achieves energy reduction while ensuring the learning accuracy. In [9], an energy-efficient bandwidth allocation and worker scheduling scheme is proposed, which minimized the energy consumption while maximizing the fraction of workers scheduled. To this end, it is also necessary to optimize the energy efficiency while investigating user scheduling schemes for implementing distributed learning algorithms in wireless networks.

To acquire the benefits of FL and SL, we propose a hybrid split and federated learning (HSFL) framework by scheduling each user with either the ST or FT method. We further investigate the energy-efficient user scheduling problem within our proposed HSFL framework, which is different from the conventional user scheduling for FL in wireless networks where the user scheduling schemes were proposed to schedule a subset of users to participate in model aggregation by considering the limited system bandwidth, the wireless channel qualities and the importance of local model updates of the users [10], [11]. Here, we have to consider one more dimension, i.e., select the appropriate training method, the FT or ST method, for the users, to achieve energy-efficient model training. The main contributions of this paper are as follows.

- We propose a novel HSFL framework by allowing the users to choose the appropriate training method, the FT or ST method, to use for participating in model training in heterogeneous wireless UAV networks.
- We formulate the user scheduling with training method selection problem in the proposed HSFL framework as an energy minimization multiple-choice knapsack problem (MCKP) [12] and propose a linear MCKP (LMCKP)-greedy user scheduling algorithm to select a subset of users in each round and schedule each user with either FT or ST method.
- The simulations demonstrate that our proposed HSFL algorithm achieves less energy consumption than FL and SFL while having the same good test accuracy performance, and the proposed LMCKP-greedy user scheduling algorithm achieves energy efficient selection of FT or ST method under different data distributions, it is especially energy saving for users under imbalanced data.

## II. System Model

We consider a wireless UAV network in which a set $\mathcal{N} = \{u_1, ..., u_N\}$ of UAVs (users) and one BS jointly train an ML model for data analysis and inference. As shown in Fig. 1, each user $u_i$ is assumed to own a local dataset $D_i$ with the data size denoted as $|D_i|$. In each round, only a subset of users is selected to participate in model training, in which each user is possible to be scheduled with the FT or ST method. For instance, in Fig. 1, $\{u_1, u_2, u_3, u_N\}$ are selected, $u_n$ is not selected because of bad channel qualities or limited energy. We consider the spatial expectation of the path loss for the line of sight (LoS) and NLoS groups as the path loss model to describe the user-BS channel as in [13].

### A. Learning Model

In this paper, we adopt DNN model to perform image recognition for fire detection in wireless UAV networks. The users and the BS collaboratively train an ML model to minimize a global loss function $F(\omega)$,

$$\min_{\omega} \; F(\omega) \triangleq \sum_{i=1}^{N} \frac{D_i}{D} F_i(\omega), \quad D = \sum_{i=1}^{N} D_i \tag{1}$$

where $D$ is the whole dataset owned by all the users, $F_i(\omega)$ is the local loss function at user $u_i$. The loss function is defined

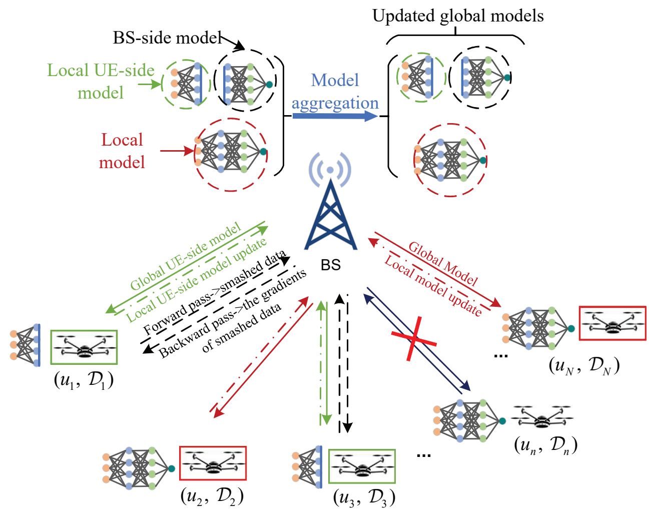



Fig. 1. The system model

according to the specific learning task, such as cross-entropy for the handwritten digit identification task.

Considering the heterogeneity of the users, we propose a new HSFL architecture to select the appropriate training method, the FT or ST method, for the users by exploiting the advantages of both SL and FL. We adopt the model splitting structure of the SL by dividing the considered DNN model into two sub-models (i.e., each sub-model contains a few NN layers) and distributing them to users and BS for collaboratively distributed training. In this case, the DNN model is trained at both the user and the BS, we adopt the parallel model training mechanism of the FL by allowing the users to perform local training at the same time. Therefore, the learning procedure of the considered DNN model with the HSFL architecture is illustrated in **Algorithm 1**.

### B. Computation Model

The whole learning procedure of the HSFL mainly contains three steps in each round: local model training of each user, the transmission of local model updates from each UE to the BS, global model aggregation, and broadcast at the BS. [1] During the local model training step, each scheduled user computes its local model updates either with the FT or ST method by using its local dataset and the received global model.

*1) FT Method:* The users scheduled with the FT method compute their local model updates with the received global model $\omega^g$ independently. Let $f_i$ be the computation capacity of user $u_i$, which is measured by the number of CPU cycles per second. Let $C_{iF}$ represent the number of CPU cycles required for computing one sample data at user $u_i$. The computation time of user $u_i$ computing its local model updates is

$$\tau_{iF}^{tr} = \frac{e_i C_{iF} D_i}{f_i}, \; \forall i \in \mathcal{K}_F, \tag{2}$$

[1] Noted that the users scheduled with the SL method perform local training by collaborating with the BS, the whole training process is defined as a local training step

**Algorithm 1** Wireless HSFL Algorithm

**Initialization** The BS initializes global model $\omega_t$, global UE-side model $\omega_t^l$ and global BS-side model $\omega_t^e$, set $t = 0$.

**Repeat**

Each user sends their characteristic information to the BS
The BS selects a subset of users $\mathcal{K}$, and schedules each selected user with the ST or FT method.
The BS distributes $\omega_t$ to the users $u_i \in \mathcal{K}_F$ with FT method, distributes $\omega_t^l$ to the users $u_i \in \mathcal{K}_S$ with ST method and at the same time assign the BS with $\omega_t^e$.
**for** $u_i \in \mathcal{K}$ in parallel **do**
  **if** $u_i \in \mathcal{K}_F$ **then**
    user computes local model updates with FT method independently.
  **else if** $u_i \in \mathcal{K}_S$ **then**
    user computes local model updates collaboratively with the BS using ST method.
  **end if**
**end for**
The BS performs model aggregation with the weighted average technique of FedAvg [4].
Set $t = t + 1$.
**Until** the desired convergence performance is achieved or the final iteration arrives

where $e_i$ is the number of local training iterations at user $u_i$. The energy consumption of computing its local model updates is $E_{iF} = \kappa e_i C_{iF} D_i f_i^2$, where $\kappa$ is the effective switched capacitance that depends on the chip architecture.

*2) ST Method:* The users scheduled with ST method compute their local model updates by collaborating with the BS, where each user calculates the local model updates of the received UE-side model and the BS calculates the local model updates of the BS-side model. Therefore, the number of CPU cycles required for computing one sample data at user $u_i$ is $C_{iS}$. We let the number of CPU cycles required for computing one sample data at the BS be $C_{iB}$, thus $C_{iS} + C_{iB} < C_{iF}$. Therefore, the computation time of user $u_i, i \in \mathcal{K}_S$ calculating the local model updates of the UE-side model and the BS-side model is

$$\tau_{iS_l}^{tr} = \frac{e_i C_{iS} D_i}{f_i}, \forall i \in \mathcal{K}_S, \ \tau_{iS_e}^{tr} = \frac{e_i C_{iB} D_i}{f}, \forall i \in \mathcal{K}_S, \quad (3)$$

The energy consumption for computing the local model updates of the UE-side model is $E_{iS} = \kappa e_i C_{iS} D_i f_i^2$

### C. Transmission Model

We assume all the users transmit their model parameters to the BS via a frequency domain multiple access (FDMA) scheme. The uplink transmission rate from the user $u_i$ to the BS is given by

$$r_i = b_i B_w log_2(1 + \frac{g_i p_i}{N_0 b_i B_w}), \quad \forall i \in \mathcal{N} \quad (4)$$

where $b_i$ is the ratio of allocated bandwidth to user $u_i$, $g_i$ is the channel gain between user $u_i$ and the BS, $B_w$ is the total bandwidth, $p_i$ is the transmit power of user $u_i$, and $N_0$ is the power spectral density of the Gaussian noise. Noted that the system bandwidth is limited, so the bandwidth can be allocated to the selected subset of users $\mathcal{K}$ should satisfy $\sum_{i \in \mathcal{K}} b_i \leq 1$.

With FT method, the user only has to upload the local model updates to the BS, where the communication overhead depends on the model size and is denoted as $m_i^g$. Then, the transmission time is $\tau_{iF}^{ul} = \frac{m_i^g}{r_i}$. Due to the high transmit power at the BS and the high bandwidth that can be used for data broadcast, the downlink transmission time is neglected.

In ST method, the UE has to upload the output activations of the cut layer and the local model updates of the UE-side model to the BS. In this case, the communication overhead includes two parts: the size of the UE-side model, denoted by $m_i^l$, and the size of the activations, denoted by $m_i^a = \boldsymbol{a} * |D_i|$, depending on the size of the local dataset $|D_i|$. Then, the uplink transmission time is $\tau_{iS}^{ul} = \frac{m_i^l + m_i^a}{r_i}$. Similarly, the UE-side model broadcasting in the downlink transmission is neglected. Therefore, the downlink transmission only includes the output gradients of the cut layer from the BS-side model, the size of the gradients is denoted by $m_i^g = \boldsymbol{g} * |D_i|$. Assuming the BS occupies the whole bandwidth to send back the gradients to each UE. The achievable downlink data rate is given by $r_s = B_s log_2(1 + \frac{p_s g}{N_s B_s})$, so the downlink transmission time is $\tau_{iS}^{dl} = \frac{m_i^g}{r_s}$.

Therefore, the one round latency of the HSFL algorithm can be written as

$$\begin{aligned} T &= \max_{i \in \mathcal{N}} (\alpha_{iF} \tau_{iF} + \alpha_{iS} \tau_{iS}), \\ \tau_{iF} &= \tau_{iF}^{tr} + \tau_{iF}^{ul}, i \in \mathcal{K}_F, \\ \tau_{iS} &= \tau_{iS}^{tr} + \tau_{iS}^{ul} + \tau_{iS}^{dl}, i \in \mathcal{K}_S, \end{aligned} \quad (5)$$

where $\alpha_{iF} = 1$ implies that the user $u_i$ is scheduled with the FT method, while $\alpha_{iS} = 1$ means the user $u_i$ is scheduled with the ST method for model training. Otherwise, $\alpha_{iF} = \alpha_{iS} = 0$ means the user $u_i$ is not selected in this round.

### D. Diversity Index

Considering the diversity of the users with different data distributions, computation capabilities and heterogeneous resources, we define a diversity index to capture the characteristic information of each user based on [14]. The diversity index is defined as the weighted sum of four parameters, including the local dataset diversity stated by the Shannon entropy [14], the user diversity indicated by the age-of-update, computation capacity and the local dataset size, denoted by $\{v_{i,1}, ..., v_{i,n}\}$ where each measure of user $u_i$ is calculated by the normalized value $v_{i,n}$. Thus, the diversity index $I_i$, is then defined as $I_i = \sum_n v_{i,n} \gamma_{i,n}$ where $\gamma_{i,n}$ is the adjustable weight for each metric $n$ of user $u_i$ assigned by the BS, it is presented as a vector $\Phi = \{\gamma_{i,1}, ..., \gamma_{i,n}\}$

## III. Problem Formulation

Due to randomly fading wireless channels and the limited energy of the users, a user scheduling scheme considering energy efficiency was investigated for implementing FL in wireless networks. By introducing the HSFL framework in the diverse wireless networks with different computation capabilities and data distributions at the users, it's essential to redesign an energy-efficient user scheduling scheme that not only chooses a subset of informative users but also determines their training methods for participating in model training in each round. We formulate this user scheduling problem as an optimization problem aiming to minimize the total energy consumption of the users and maximizing the diversity of the users under the latency constraint. Here, we only consider the energy consumption of the users includes both local computation energy and communication energy. Noted that more diverse users are preferred in each round, so a negative sign is added to the diversity coefficient here, and each user can select either FT or ST method, denoted by a set $\mathcal{J} = \{F, S\}$. The formulated optimization problem is presented as

$$
\begin{aligned}
\mathcal{OP}_1 \quad & \min_{b_i,\,\{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij}(E_{ij} - I_{ij}) \\
s.t. \quad & C_1: \sum_{j \in \mathcal{J}} \alpha_{ij}\tau_{ij} \le T, \\
& C_2: \alpha_{ij} \in \{0, 1\},\ \forall j \in \mathcal{J},\ \forall i \in \mathcal{N}, \\
& C_3: \sum_{j \in \mathcal{J}} \alpha_{ij} \le 1,\ \forall i \in \mathcal{N}, \\
& C_4: \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij} b_i \le 1, \\
& C_5: 0 \le b_i \le 1,\ \forall i \in \mathcal{N},
\end{aligned}
\tag{6}
$$

where $C_1$ indicates there is a maximum constraint for one round latency. $C_2$ is the constraint of the binary variable $\alpha_{ij}$ that indicates if the user $u_i$ is scheduled with the $j$ training method or not. $C_3$ indicates each user $u_i$ can only be scheduled with maximum one training method $j$. $C_4$ is the constraint that the sum of the bandwidth allocated to all the users cannot exceed the total bandwidth $B_w$, and $C_5$ gives the values of the bandwidth allocation ratio. The optimization problem $\mathcal{OP}_1$ is a mixed integer multi-objective problem that is non-linear and hard to be solved with direct mathematical tools. Therefore, we transform our formulated problem in (6) to an MCKP problem which is an extension of the knapsack problem [12].

To solve the formulated problem (6), we first formulate it as a standard MCKP problem as $\mathcal{OP}_2$ in (7) without considering the time constraint $C_1$, and then we obtain the feasible solution by solving it with the proposed LMCKP-greedy algorithm. At last, we search the final solution under $C_1$ within the obtained feasible solution. Thus, the standard MCKP problem and the time constraint are written as follows,

$$
\begin{aligned}
\mathcal{OP}_2 \quad & \min_{\{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij}(E_{ij} - I_{ij}) \\
s.t. \quad & C_2,\ C_3,\ C_4,\ C_5
\end{aligned}
\tag{7}
$$

$$
\sum_{j \in \mathcal{J}} \alpha_{ij}\tau_{ij} \le T,\quad \forall i \in \mathcal{N} \tag{8}
$$

The formulated user scheduling problem with multiple training methods selection can be perfectly mapped to the MCKP problem, in which one training method should be selected for each user $u_i, i \in \mathcal{N}$ such that the total energy consumption of the users is minimized and the user diversity is maximized, while the overall bandwidth does not exceed the available bandwidth. In the MCKP-based user scheduling problem, the users and the training methods are mapped to the sets of items $S_i,\ i \in \mathcal{N}$ and the items in each set, respectively. Thus, the $S_i$ represents the set of training methods $m_j, j \in \{F, S\}$ for the user $u_i$. Noted that the users may not be successfully selected due to serious channel fading or restricted energy resource, in this case, we add the possibility of the user not being selected as a new training method with $m_j = 0, j = 0$ in the set of training methods $S_i, i \in \mathcal{N}$. Therefore, $S_i$ is updated as $S_i = \{m_j, j \in \mathcal{J},\},\ \mathcal{J} = \{0, F, S\}$.

The weighted sum of energy consumption and user diversity, and the bandwidth allocation ratio of the user $u_i$ scheduled with the training method $m_j$ are mapped to the profit and weight of the item $m_j$ in the set $S_i$, respectively. Here, the bandwidth allocation ratio is straightly used as the weight, while the profit is defined as $\mathbf{p}_{ij} = C + (I_{ij} - E_{ij})$, where $C$ is a large constant number ensuring that $\mathbf{p}_{ij}$ is a non-negative number. As a result, the optimization problem for user scheduling is reformulated as a standard MCKP problem as follows

$$
\begin{aligned}
\mathcal{OP}_3 \quad & \max_{\{\alpha_{ij}\}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}} \alpha_{ij} p_{ij} \\
s.t. \quad & C_2,\ C_4, \\
& C_3: \sum_{j \in \mathcal{J}} \alpha_{ij} = 1,\ \forall i \in \mathcal{N}
\end{aligned}
\tag{9}
$$

## IV. LMCKP-greedy User Scheduling Algorithm

In this section, we propose a user scheduling algorithm to solve the formulated MCKP optimization problem. It is impossible to solve the formulated MCKP-based user scheduling problem for a practical application since it is NP-hard. Hence, we relax the constraint $C_4$ in $\mathcal{OP}_3$ to a linear constraint,

$$
0 \le \alpha_{ij} \le 1,\ i \in \mathcal{N},\ j \in \mathcal{J}. \tag{10}
$$

Then, the linear MCKP (LMCKP) problem can be solved in O($nlogn$) time through the following two steps: 1) remove the dominated training mechanisms to reduce the scale of the given LMCKP and 2) apply the greedy algorithm.

### A. Removing Dominated Training Mechanisms

Given the user $u_i$, the dominated training methods can be removed because a more profitable training method always exists. This means a training method providing a higher profit and having a lighter weight is always preferred as the training method for a given user $u_i$ as an optimal solution. In LMCKP, the training method $m_{is}$ is considered to be dominated by the training method $m_{ir}$ and $m_{it}$ if the three methods satisfy the following conditions

$$w_{ir} \leq w_{is} \leq w_{it},\ p_{ir} \leq p_{is} \leq p_{it}, \tag{11a}$$

$$\lambda_{i,r\to s} \leq \lambda_{i,r\to t}, \tag{11b}$$

$$\lambda_{i,r\to t} = \frac{p_{it} - p_{ir}}{w_{it} - w_{ir}},\ \lambda_{i,r\to s} = \frac{p_{is} - p_{ir}}{w_{is} - w_{ir}}, \tag{11c}$$

where $\lambda_{i,r\to t}$ is defined as the update efficiency, which can be interpreted as the normalized profit increasing with respect to the weight increment. Here, (11b) means switching from $m_{ir}$ to $m_{is}$ is less efficient than switching from $m_{ir}$ to $m_{it}$.

### B. The Greedy Algorithm

After removing the dominated training methods, only the remaining training methods are considered as candidates for the solution to the LMCKP problem. Then, we can follow the greedy algorithm to obtain the LMCKP solution. First, we initialize the training method with the smallest bandwidth ratio for each user and set $\alpha_{i1} = 1$. Second, we compute and sort the update efficiency of each training method in non-decreasing order, and update the training method of each user from $m_{ij}$ to $m_{ik}$ by letting the corresponding selection variable $\alpha_{ij}$ and $\alpha_{ik}$ to 0 and 1, respectively. Moreover, the total bandwidth ratio of all the users is updated due to the change of training method at each user, $b = b - b_{ij} + b_{ik}$. The second step is repeated until the constraint $C_4$ is not satisfied.

The final step is to determine the MCKP-feasible solution from the solution of the LMCKP problem, in which the selection variables $\alpha_{ij}, i \in \mathcal{N}, j \in \mathcal{J}$ are either 0 or 1 to satisfy the constraint $C_2$. The MCKP-feasible solution determines the selected training method for each user. In the second step, if $b = 1$, then the obtained solution is the optimal solution to both LMCKP and MCKP, and the repeated process is terminated. Otherwise, only the optimal solution to LMCKP is obtained, where the selection variables of the last training method updating $\alpha_{ij}$ and $\alpha_{ik}$ are fraction numbers. To obtain MCKP feasible solution, the last update is reversed by resetting the selection variables $\alpha_{ij} = 1$ and $\alpha_{ik} = 0$. Then, the MCKP-feasible solution is obtained as $\mathcal{N}_{candi}$ with the scheduled training method for each user.

Next, we obtain the final solution by searching for the solutions that satisfy the constraint (8). Therefore, the detailed steps of the proposed LMCKP-greedy user scheduling algorithm is summarized in **Algorithm 2**.

**Algorithm 2** LMCKP-greedy User Scheduling Algorithm

**Initialization:**

**Input:** The diversity index $I_i, i \in \mathcal{N}$, energy consumption of each learning mode $E_{ij}, i \in \mathcal{N}, j \in \mathcal{J}$

Set $T_{th} = 10$

**Learning:**

```
for t ≤ T do
    The profit is calculated as p_ij, the weight matrix is
    initialized as b_ij = 1/N, if j = {S, F}, otherwise b_ij = 0.
    N_candi = LMCKPgreedy (profit, weight)
    for u_i ∈ N_candi do
        if u_i ∈ K_F and τ_i ≤ T_th then
            update local model updates with FL method.
        else if u_i ∈ K_S and τ_i ≤ T_th then
            update local model updates with SL method.
        end if
    end for
    The BS perfroms model aggregation with the received
    local model updates.
end for
```

***LMCKPgreedy:***

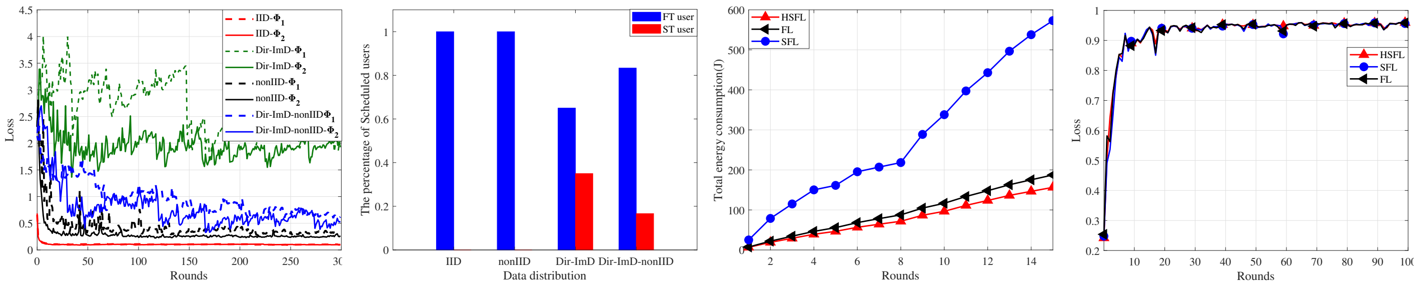
```
Remove dominated training mechanisms according to (11),
set α_ij = 1
Sort λ_{i,r→t} in non-decreasing order, and update the selec-
tion variable α_ij for each training mechanism; then update
b = b − b_ij + b_ik; repeat until violates the constraint C_4.
Obtain the MCKP feasible solution N_candi.
Return N_candi .
```

## V. Simulation Results

In this section, we conduct the experiments of image recognition on MINST dataset to illustrate the learning performance of our developed user scheduling scheme with the proposed HSFL framework. A DNN model with two convolution layers and two fully connected layers is considered, which uses $5 \times 5$ sized kernels in the convolution layers. A wireless UAV network with one BS located at the origin of the cell and multiple UAVs uniformly distributed within the cell is simulated. The cell radius is 500 m, the height of the BS antenna is 20 m, and the flying height of UAVs is set from the range of $(20-80)$ m. The detailed simulation parameters of the UAV networks are provided in Table I.

We simulate four different data distributions over $N = 100$ users to imitate the practical scenarios. The IID and nonIID data distribution follow the settings in [4]. We use Dirichlet distribution [15] to set the Dirichlet imbalanced data distribution (Dir-ImD) with Dir ($\alpha_d = 0.1, \alpha_{imd} = 2$), and the Dirichlet nonIID and imbalanced data distribution (Dir-nonIID-ImD) with Dir ($\alpha_d = 0.01, \alpha_{imd} = 2$). The smaller $\alpha_d$ indicates larger data heterogeneity across users and smaller $\alpha_{imd}$ indicates the dataset size across users is more imbalanced. We set the local training rounds $e_i = 5$ and the batch size $b = 10$. Let the maximum one round latency be $T_{th} = 10$, the computation capacity of each user $f_i$ randomly choose from $\{0.1\text{GHz}, 1\text{GHz}, 2\text{GHz}, 3\text{GHz}\}$, $f = 8$ GHz. We set $C_{iF} = 3 \times 10^4$, $C_{iS} = 5 \times 10^3$, $C_{iB} = 2.5 \times 10^4$, and $\kappa = 10^{-28}$.

The convergence performance of our proposed HSFL framework with MCKP-based user scheduling scheme under IID, nonIID, Dir-ImD, and Dir-ImD-nonIID data distributions is shown in Fig. 2 (a). The weight vector of the diversity index $\Phi$ has an impact on the convergence performance of the model training, where $\Phi_1 = [0.25, 0.25, 0.25, 0.25]$, $\Phi_2 = [0.4, 0.1, 0.1, 0.4]$, we can observe that the curves with $\Phi = \Phi_2$ achieve better convergence performance, which indicates that large local model updates and diverse users are more important




(a) Convergence performance (b) Scheduled user percentage (c) Energy consumption comparison (d) Test accuracy

Fig. 2. (a,b): The comparisons of convergence performance and user scheduling performance with the HSFL framework under IID, nonIID, Dir-ImD and Dir-ImD-nonIID data distributions. (c,d) Energy consumption and test accuracy comparisons among HSFL, FL and SFL under Dir-ImD data distribution.

TABLE I
SIMULATION PARAMETERS OF WIRELESS UAV NETWORKS

| Parameters | Value |
|---|---|
| $\varphi_l, \varphi_n$ | 21, 1 |
| $a, b$ | 5.0188, 0.3511 |
| Rician factor Kdb | 2 dB |
| system carrier frequency | 2 GHz |
| Noise power $\sigma^2$ | −130 dBm |
| $P_s, P_n$ | 40 dBm, 24 dBm |
| $B_s, B_w$ | 5 MHz, 1 MHz |

to the convergence of model training. In Fig. 2 (b), we plot the percentages of scheduled users with the FT or ST method under different data distributions. It is shown that the users tend to choose the ST method under ImD data distribution. This is because the users owning small local datasets uses the ST method instead of the FT method can save more energy.

Fig. 2 (c) plots the energy consumption comparisons among FL, SFL, and our proposed HSFL algorithms under Dir-ImD data distribution. It is shown that our proposed HSFL consumes the lowest energy but achieves the same test accuracy compared to FL and SFL as shown in Fig. 2 (c) and (d). This is mainly because the communication overhead is reduced by selecting the ST method for the users with a relatively smaller dataset while selecting the FT method for the users with a larger dataset to transmit fewer model parameters, thus the energy is saved especially under the ImD data. Moreover, our proposed HSFL adopts a similar parallel training mechanism and model aggregation rule as FL and SFL, so it can achieve the same training results.

## VI. CONCLUSION

In this paper, we proposed a new hybrid split and federated learning (HSFL) framework for distributed model training in wireless networks, taking into account the diversity of the users with diverse computational capabilities and data distributions. Considering the randomly fading channels and limited energy at the users, we formulated the user scheduling problem as an energy minimization multiple-choice knapsack problem (MCKP), and developed an energy-efficient linear MCKP (LMCKP)-greedy user scheduling algorithm to select a subset of users for model training in each round and schedule each user with either split training (ST) or federated training (FT) methods. Our results demonstrated the feasibility of the user scheduling algorithm under IID, non-IID, and ImD data distributions and it could choose the ST method under ImD data to save energy. Our proposed HSFL is shown to consume less energy than FL and SFL but achieves the same good test accuracy performance under ImD data distribution.

## REFERENCES

[1] U. Challita, W. Saad, and et al, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Mar. 2019.

[2] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[3] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53 841–53 849, Mar. 2020.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, Apr. 2017, pp. 1273–1282.

[5] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, Dec. 2018.

[6] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv preprint arXiv:1909.09145*, Sep. 2019.

[7] C. Thapa, M. A. P. Chamikara, and S. Camtepe, "Splitfed: When federated learning meets split learning," *arXiv preprint arXiv:2004.12088*, Sep. 2021.

[8] Z. Yang, M. Chen, and et al., "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2020.

[9] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops*, Jun. 2020, pp. 1–6.

[10] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun.(ICC), Shanghai,*, May 2019, pp. 1–7.

[11] M. M. Amiri, D. Gündüz, and et al., "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, Jan. 2021.

[12] D. Pisinger, "A minimal algorithm for the multiple-choice knapsack problem," *Eur. J. Oper. Res.*, vol. 83, no. 2, pp. 394–410, Jun. 1995.

[13] J. H. et al, "Elevation dependent shadowing model for mobile communications via high altitude platforms in built-up areas," *IEEE Trans. on Ant. and Prop.*, vol. 56, no. 4, pp. 1078–1084, Apr. 2008.

[14] A. Taik, Z. Mlika, and S. Cherkaoui, "Data-aware device scheduling for federated edge learning," *arXiv preprint arXiv:2102.09491*, Feb. 2021.

[15] 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*, Sep. 2019.