Energy-Efficient Decentralized Federated Learning for UAV Swarm with Spiking Neural Networks and Leader Election Mechanism

Chen Shang, Dinh Thai Hoang, *Senior Member, IEEE,* Min Hao, Dusit Niyato, *Fellow, IEEE,*,
Jiadong Yu, *Member, IEEE,*

***Abstract*—Federated Learning (FL) has been considered a critical technique for assisting Unmanned Aerial Vehicle (UAV) swarm to efficiently perform tasks in dynamic environments. However, deploying FL in UAV swarm is constrained by the limited energy of the UAVs and the complex communication environments within UAV swarm networks. This work introduces a leader election-assisted Spiking Neural Networks (SNNs)-driven decentralized FL framework for UAV swarm. This framework enables UAV swarm to train a high-performance FL model while minimizing energy and time consumption, thereby enhancing real-time decision ability of UAV swarm. In particular, the SNN-driven FL allows UAV swarm to train a shared model with less energy consumption through its discrete spike event. To this end, we conduct a systematic analysis of the training challenges associated with SNN-driven FL, and we then propose an approximate derivative algorithm to address these challenges. Furthermore, we develop an intelligent leader selection scheme based on Bayes theorem designed to reduce time consumption of model parameter transmission and accelerate the model aggregation. Simulation results show that the proposed scheme outperforms baseline schemes in terms of model performance, energy and time consumption.**

***Index Terms*—Unmanned Aerial Vehicle, Federated learning, Spiking Neural Network**

I. Introduction

Unmanned Aerial Vehicle (UAV) swarms represent a groundbreaking advancement in technology, offering a multitude of practical applications. These swarms consist of multiple coordinated UAVs, leverage advanced algorithms to mimic natural swarm behaviors, enhancing their efficiency in complex tasks like disaster management and environmental monitoring [1]. The emergent behavior of UAV swarms introduces new opportunities in automation and artificial intelligence, marking it as a rapidly evolving field. Federated Learning (FL) is a revolutionary machine learning approach that allows multiple UAVs in a swarm to collaboratively learn a shared model while keeping all the training data on-device. In practice, FL can enhance the operational efficiency of UAV swarms in tasks like synchronized flight patterns, obstacle avoidance, and area surveillance, by enabling on-the-fly learning and decision-making, which is crucial in dynamic and unpredictable environments [2].

C. Shang and J. Yu are with the Internet of Things Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China (chenshang@hkust-gz.edu.cn, jiadongyu@hkust-gz.edu.cn). M. Hao is with the Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore (min_hao@sutd.edu.sg). D. T. Hoang is with the School of Electrical and Data Engineering, University of Technology Sydney, Australia (hoang.dinh@uts.edu.au). Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (dniyato@ntu.edu.sg)

Despite the advantages offered by FL, its effectiveness in UAV swarm is hampered by the inherent characteristic of UAV, e.g., limited energy and higher mobility. The finite energy resources of UAVs, predominantly allocated to maintain flight, impose strict limitations on their computational capacities, significantly influencing the extent, complexity, effectiveness of the data processing, and learning tasks that can be undertaken by UAVs. Moreover, the high mobility of UAVs leads to dynamic changes in network topology, posing challenges in achieving stable connectivity and managing latency. These issues critically affect the communication and synchronization necessary for effective FL [3].

Recently, several approaches have been proposed to address above challenges. In [4], an energy-efficient dynamic scheduling scheme was proposed, optimizing FL performance by selecting specific UAVs to participate in model training. The authors of [5] analyzed how the limited energy and high mobility of UAVs impact FL convergence and designed a framework to address these challenges. [6] explored how to achieve high-level FL performance in a UAV swarm by jointly optimizing training and network resources. Although these methods effectively address the negative impacts of limited energy and high mobility on FL performance, primarily focusing on optimizing energy allocation, none of them considered how the model architecture of FL affects energy consumption within the UAV swarm. Additionally, [4] utilized over-the-air computation (AirComp) for aggregating FL models. However, the requirement for strictly synchronized transmissions in AirComp poses challenges in highly mobile UAV swarms.

This work aims to propose a novel framework leveraging the advantages of Spiking Neural Networks (SNNs) together with an intelligent leader election mechanism to address the above problems for FL-based UAV swarms. SNNs, recognized as the third generation of neural networks, present a promising approach to significantly reduce energy consumption for both training models and tasks inference using their bio-inspired architecture. Different from traditional neural networks (e.g., Artificial Neural Networks (ANNs)) that process information through continuous values, SNNs leverage discrete events (i.e., spike) for information transmission. This method allows SNNs to activate and process information only in response to spikes, substantially reducing unnecessary computational activities and thus conserving energy. To enhance the efficacy of SNN-driven FL deployed in UAV swarms, we conduct a systematic analysis of the training challenges associated with SNNs. Subsequently, we introduce an approximate derivative algorithm to address these challenges, enabling the UAV

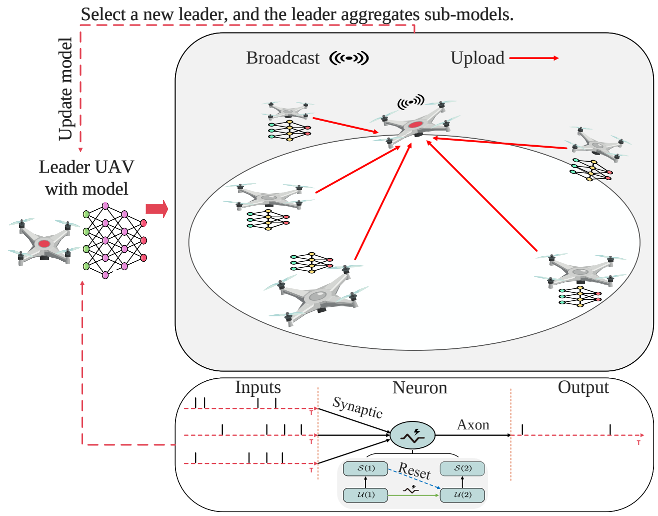

Fig. 1: An illustration of the SNN-Driven decentralized FL framework. The leader UAV broadcasts the model to other UAVs for local training. After completing local training, all model parameters are transmitted to a new leader UAV, which then aggregates all received model parameters to update the model.

swarm to train high-performance FL models while maintaining lower energy consumption. Finally, we develop a decentralized leader election approach leveraging the Bayes theorem to mitigate the challenges posed by the high mobility of UAVs. To the best of our knowledge, this is the first work using SNN-driven FL model with leader election mechanism for UAV swarm. The simulation results show that our proposed framework can outperform other baseline schemes in terms of model performance, energy and time consumption.

II. System Model

As shown in Fig. 1, we consider a UAV swarm, each UAV represented by $\mathcal{K} = \{1, 2, \ldots, K\}$. All UAVs cooperatively execute an FL algorithm over wireless networks. In FL, the phase where each UAV trains its local model using its own collected data is named as local training, and the process where these local models are transmitted to the leader UAV for aggregation is called the global learning round.

A. Federated Learning Model

Let $F_k(\boldsymbol{w}) = \mathcal{L}(\boldsymbol{w}, \mathcal{D}_k)$ represent the local training loss of UAV-k with respect to model weight $\boldsymbol{w}$, where $\mathcal{L}(\cdot,\cdot)$ is the predetermined loss function. The objective of the FL task is to minimize the specified global loss function

$$F(\boldsymbol{w}) = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{\mathcal{D}} F_k(\boldsymbol{w}). \tag{1}$$

As shown in Fig. 1, we propose a new FL-based UAV swarm framework with two techniques to reduce the energy and time consumption, i.e., SNN-driven FL and leader selection mechanism. Under the coordination of multiple UAVs, the UAV swarm cyclically executes the following steps: (1) Selecting the leader UAV; (2) The leader UAV broadcasts the latest global model $\boldsymbol{w}$ to other UAVs; (3) Each UAV computes and updates its local gradients $\boldsymbol{w}_k$ using its local data; (4) the UAVs upload new models to the leader UAV.

B. Computation and Communication Model

We use $\mathcal{D}_k$ with size $|\mathcal{D}_k|$ to denote the local training data of the UAV-k, and $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ is the global data.

1) Computation Model: For the UAV-k at the e-th global learning round, given the computing capacity of UAV-k $f_{e,k}$, the local training time can be calculated as [7]

$$T_{e,k}^{\text{cmp}} = \frac{\tau |\mathcal{D}_k| C_k}{f_{e,k}}, \tag{2}$$

where τ is the number of local training rounds, and C_k (cycles/bit) denotes the number of Central Processing Unit (CPU) cycles needed for computing one data sample of UAV-k. The energy consumption for computing at UAV-k can be calculated as [8]

$$E_{e,k}^{\text{cmp}} = \varphi f_{e,k}^2 \tau |\mathcal{D}_k| C_k, \tag{3}$$

where φ denotes the effective switched capacitance. For the leader UAV, the additional computation time and energy consumption for model aggregation per global epoch is computed by [9]

$$\begin{aligned} T_{e,\text{leader}}^{\text{Ag}} &= \frac{\sum_{k=1}^{K} |\mathcal{M}_k| C_{\text{leader}}}{f_{e,\text{leader}}}, \\ E_{e,\text{leader}}^{\text{Ag}} &= \varphi f_{e,k}^2 \tau \sum_{k=1}^{K} |\mathcal{M}_k| C_{\text{leader}}, \end{aligned} \tag{4}$$

where $|\mathcal{M}_k|$ denotes the data size of the model update.

2) Communication Model: All UAVs will upload their local FL models to the leader UAV through PC5 interface (i.e., the direct communication mode without the help of a base station) [10] after local computation. In this work, we consider an orthogonal frequency division multiple access (OFDMA) scheme with a total bandwidth of B for wireless transmission. At the e-th global learning round, for the UAV-k allocated with the bandwidth b_k, the achievable transmission rate between the UAV-k and the UAV-q ($k \neq q$) can be expressed as

$$r_{k,q} = b_k \log_2\left(1 + \frac{P_k g_{k,q}}{\delta b_k}\right), \tag{5}$$

where P_k is the average transmit power of UAV-k, $g_{k,q}$ is the channel gain between UAV-k and UAV-q, and δ denotes the power spectral density of the Gaussian noise.

For the UAV-k at e-th global learning round, in order to upload and broadcast updated model $\mathcal{M}_k$ with data size $|\mathcal{M}_k|$, the communication time $T_{\text{e},k}^{\text{com}}$ (i.e., upload time $T_{\text{e},k}^{\text{up}}$ and broadcast time $T_{\text{e},k}^{\text{bc}}$) and energy consumption E_k^{com} can be respectively expressed as

$$T_{\text{e},k}^{\text{com}} = T_{\text{e},k}^{\text{up}} + T_{\text{e},k}^{\text{down}}, E_{\text{e},k}^{\text{com}} = T_{\text{e},k}^{\text{com}} P_k, \tag{6}$$

where upload and broadcast time are equal to the model size $|\mathcal{M}_k|$ divided by the current achievable transmission rate r_k. Note that we ignore the receiver power consumption, which allows us to calculate the energy consumption for uploading and broadcasting using $E_{e,k}^{\text{com}}$ [7]–[9].

Therefore, for the e-th global learning round, all the training time is given by

$$T_e^{total} = \tau\left\{\max_k T_{e,k}^{\text{cmp}} + \max_k T_{e,k}^{com} + \max_k T_{e,\text{leader}}^{\text{Ag}}\right\}. \tag{7}$$

We use the synchronized updates scheme [8], wherein the leader UAV aggregates the global model only after receiving all local models from the other UAVs.

C. Energy Consumption Model of UAV

The flight energy consumption E_k^{flt} of UAV-k is denoted by the UAV-k propulsion energy consumption per unit travelling distance in Joule/meter (J/m) with speed v_k, which can be expressed as [11]

$$E_k^{\text{flight}}\left(T^{total}, v_k\right) = \int_0^{T^{total}} \mathcal{P}\left(v_k\right) dt, \tag{8}$$

where $\mathcal{P}\left(v_k\right)$ is the power of flight. For a global learning round, the total energy consumption of UAV-k can be expressed as

$$E_{e,k}^{total} = \begin{cases} E_{e,k}^{\text{cmp}} + E_{e,\text{leader}}^{\text{Ag}} + E_{e,k}^{\text{flight}} + E_{e,k}^{\text{com}}, k \text{ is leader} \\ E_{e,k}^{\text{cmp}} + E_{e,k}^{\text{com}} + E_{e,k}^{\text{flight}}, \text{otherwise} \end{cases}. \tag{9}$$

III. Spiking Neural Networks-Driven FL and Leader Selection Based on Bayes Theorem

In this section, we first introduce the SNNs-driven FL and its training challenges, and then we propose an approximate derivative algorithm to solve these challenges. As aforementioned, the SNNs can reduce energy consumption through its characteristic of spike event. Therefore, we consider only how to efficiently train an SNN model in this section, the specifics of conserving energy are detailed in Section IV. Moreover, we propose a leader selection mechanism for accelerating the convergence time of FL. Specifically, different from traditional FL framework where all edge devices transmit their local models to the central server, our proposed leader election mechanism enables UAV swarms to operate efficiently without a central server (i.e., decentralization), thereby expanding their application scenarios (e.g., areas without a base station) and enhancing their flexibility. Foremost, the leader election mechanism significantly reduces the transmission time of models since all model parameters are transmitted to the leader UAV, which is typically closer than a central server.

A. Spiking Neural Networks-Driven FL

We use the SNNs architecture to build an FL model, as shown in Fig. 1. The spiking neurons are modeled by the Leaky-Integrate-and-Fire (LIF) variant mechanism [12], which can be expressed as

$$\mathcal{U}_j(t+1) = (1-\lambda)\mathcal{U}_j(t) + \lambda \sum_{i=1}^{N} W_{i,j}\mathcal{I}_i(t), \tag{10}$$

where $\mathcal{U}_j\left(t\right)$ represents the membrane potential of the j-th neuron at time step t, λ is the constant of membrane potential decay with time, N is the set of neurons connected to neuron j, $W_{i,j}$ and $\mathcal{I}_i\left(t\right)$ denote the synaptic weight and input potential from upstream neuron i, respectively. The neuron j fires a spike $\mathcal{S}_j\left(t\right)$ if its membrane potential $\mathcal{U}_j\left(t\right)$ surpasses the firing threshold $\mathcal{U}_{\text{th}}$, and then, the membrane potential $\mathcal{U}_j\left(t\right)$ goes back to a reset value $\mathcal{U}^r\left(t\right)\left(\mathcal{U}^r\left(t\right) < \mathcal{U}_{\text{th}}\right)$. The firing process of spike is given by

$$\mathcal{S}_j = \begin{cases} 1, & \text{if } \mathcal{U}_j(t) \geq \mathcal{U}_{\text{th}} \\ 0, & \text{otherwise} \end{cases}. \tag{11}$$

B. Training Analysis for SNN-Driven FL Model

Neural networks learn and improve their accuracy by iteratively adjusting internal parameters. This process requires the loss function of model to be differentiable, which enables the efficacy of back propagation. In SNNs, the back propagation process can be expressed as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{i,j}} &= \sum_{t=1}^{T-1} \frac{\partial \mathcal{L}}{\partial \mathcal{U}_j\left(t\right)} \frac{\partial \mathcal{U}_j\left(t\right)}{\partial W_{i,j}} + \frac{\partial \mathcal{L}}{\partial \mathcal{U}_j\left(T\right)} \frac{\partial \mathcal{U}_j\left(T\right)}{\partial W_{i,j}} \\ &= \sum_{t=1}^{T-1} \left(\frac{\partial \mathcal{L}}{\partial \mathcal{S}_j\left(t\right)} \frac{\partial \mathcal{S}_j\left(t\right)}{\partial \mathcal{U}_j(t)} + \frac{\partial \mathcal{L}}{\partial \mathcal{U}_j(t+1)} \frac{\partial \mathcal{U}_j(t+1)}{\partial \mathcal{U}_j(t)} \right), \\ &\quad \times \frac{\partial \mathcal{U}_j(t)}{\partial W_{i,j}} + \frac{\partial \mathcal{L}}{\partial \mathcal{S}_j\left(T\right)} \frac{\partial \mathcal{S}_j\left(T\right)}{\partial \mathcal{U}_j(T)} \frac{\partial \mathcal{U}_j(T)}{\partial W_{i,j}} \end{aligned} \tag{12}$$

where T denotes the time step of SNNs training and will be detailed in Section IV. It can be found that (12) contains $\frac{\partial \mathcal{L}}{\partial \mathcal{S}_j(t)}$ and $\frac{\partial \mathcal{S}_j(t)}{\partial \mathcal{U}_j(t)}$. Specifically, the derivative of $\mathcal{S}_j$ is impulse function, which is defined by

$$\delta\left(x\right) = \begin{cases} +\infty, x = 0 \\ 0, x \neq 0 \end{cases}, \tag{13}$$

therefore, it is unstable and non-trivial to directly calculate the gradient and apply the gradient descent algorithm in SNNs.

C. Approximate Derivative Algorithm for SNNs Training

Based on the analysis in Section III-B, we propose an approximate derivative algorithm to address the non-differentiable part of SNN-driven FL, which enables UAV swarm to efficiently update their model gradients and complete learning task. Note that the algorithm is used only for back propagation, the forward propagation still follows (10) and (11). Define the curve function $f(\mathcal{U}-\mathcal{U}_{\text{th}})$ to approximate the spike output function $\mathcal{S}\left(\mathcal{U}-\mathcal{U}_{\text{th}}\right)$, which can guarantee that $f(\mathcal{U}-\mathcal{U}_{\text{th}})$ is well-defined and has a very close value as $\mathcal{S}\left(\mathcal{U}-\mathcal{U}_{\text{th}}\right)$ when $\mathcal{U}-\mathcal{U}_{\text{th}} = 0$. The function $f(\mathcal{U}-\mathcal{U}_{\text{th}})$ is defined as [12]

$$f\left(\mathcal{U}-\mathcal{U}_{\text{th}}\right) = \frac{1}{\pi}\arctan\left(\frac{\pi\eta}{2}\left(\mathcal{U}-\mathcal{U}_{\text{th}}\right)\right) + \frac{1}{2}, \tag{14}$$

where $\eta > 0$ is the custom parameter. Let $\mathcal{U}-\mathcal{U}_{\text{th}} = x$, the derivative of $f(x)$ is

$$f'\left(x\right) = \frac{\eta}{2} \times \frac{1}{1+\left(\frac{\pi\eta}{2}x\right)^2}. \tag{15}$$

(14) and (15) enable SNN to acquire meaningful gradient while SNN utilizes gradient descent algorithm. Therefore, the function f can be approximated as $\lim\limits_{x\to 0} f\left(x\right) \approx \lim\limits_{x\to 0} \mathcal{S}\left(x\right)$. In (12), $\frac{\partial \mathcal{L}}{\partial \mathcal{S}_j(t)}$ and $\frac{\partial \mathcal{S}_j(t)}{\partial \mathcal{U}_j(t)}$ can be replaced by the approximate derivative function $f\left(\cdot\right)$.

D. Leader Selection Based on Bayes Theorem

As explained above, leader election significantly influences performance of the model aggregation. Electing a leader with robust communication ability and sufficient energy can accelerate the FL model aggregation, thereby reducing the total training time. Conversely, if a UAV with limited energy and communication capability is chosen as the leader, it can adversely affect the latency of the model aggregation.

Define a prior probability $\Pr\left(L_k\right)$ to represent the likelihood of UAV-k being selected as the leader before considering its current state, which can be initialized by setting uniformly across all UAVs. The likelihood function, defined as the probability of observing the current state θ (i.e., communication ability and energy consumption) given that UAV-k is the leader [13], can be expressed as

$$\Pr\left(\theta\left|L_k\right.\right)=\frac{1}{1+\exp\left(-\left(\lambda_c\bar{r}_k-\lambda_e E_k^{total}\right)\right)},\tag{16}$$

where $\bar{r}_k=\sum_q^K r_{k,q}/K$ is the average communication quality between UAV-k and other UAVs, E_k^{total} represents the energy consumption. λ_e and λ_c are regulatory factors. Then, the posterior probability, defined as the probability of UAV-k being selected as the leader UAV based on its current state, is given by

$$\Pr\left(L_k\left|\theta\right.\right)=\frac{\Pr\left(\theta\left|L_k\right.\right)\times\Pr\left(L_k\right)}{\sum_{k=1}^K\Pr\left(\theta\left|L_k\right.\right)\times\Pr\left(L_k\right)}.\tag{17}$$

At the e-th global learning round, all UAVs update the likelihood function and calculate posterior probability based on (16) and (17), respectively. The UAV with the highest posterior probability is elected as the leader. Besides, this posterior probability is then leveraged as the prior probability for the $e+1$-th global learning round. This method enables UAV with the robust communication capabilities and lower energy consumption to be selected as the leader in each global learning round, thereby enhancing the efficacy of the FL model.

IV. Performance Evaluations

A. Simulation Settings

1) UAV swarm: We consider $K=20$ UAVs randomly flight within 1 km $\times$ 1 km square region for collecting data and performing the FL training process. All the UAVs fly with a fixed speed $v\sim[10,15]$ m/s. The path loss model is represented as $128.1+37.6\log_{10}d_k$, and d_k denotes the distance between the UAV k and the leader UAV in kilometers [8]. Besides, the noise power spectral density is $\delta=-174$ dBm/Hz [8], and the average transmit power and bandwidth are $P_k=20$ dB and $b_k=1$ MHz for all UAVs [8], respectively. Each UAV has an equal computation capacity f_k = 2 GHz, C_k = 20 cycles/bit [9]. In addition, the effective switched capacitance is $\varphi=10^{-28}$. Moreover, all flight-related parameters are set the same as [11].

2) SNN setup: The SNN includes a spiking encoder network and a classifier network. The spiking encoder network is composed of $2\times\{Conv2d-BatchNorm-LIF-Pool\}$. The spiking encoder can extract features from inputs and convert them into the firing spikes at different time-steps. We take the average-pooling for maintaining the smoothness and continuity of signals and ensuring the effective capture of global spike features across the entire input. The classifier network consists of $2\times\{Dropout-FC-LIF\}$. Specifically, the output of the spiking neuron is binary, and the direct use of the results of a single run for classification is easily disturbed by the noise caused by coding. Therefore, the output of the SNN can be regarded as the release frequency of the output layer within a period of time, and the level of the release rate indicates the response size of this category [12]. As a result, the network necessitates operation across a span of time, leveraging the average spiking rate observed after T moments as the foundational criterion for classification. In this letter, we set $T=6$. Besides, the LIF related parameters $\eta=3$, $\mathcal{U}_{\rm th}=1$, $\mathcal{U}^r=0$, and $\lambda=\frac{1}{2}$ [12].

3) FL training: We use Fashion-MNIST [14] and MNIST [15] to validate our proposed method, all datasets with data size 44.92 MB are first shuffled and then partitioned into $60,000/20=3000$ equal parts. We employ the model as illustrated in *SNN setup*, with a model update data size of 3.29 MB, i.e., $|\mathcal{M}_k|=3.29$ MB. For the learning hyper-parameters, the learning rate, batch size and local epoch are set as $\{0.01,64,1\}$. Besides, the $\lambda_c=1$ and $\lambda_e=1$, respectively.

4) Baselines: We conduct performance comparisons between our proposed method, denoted by SFLwE, and several baselines (i) SNNs-driven FL without leader election (SFLE), (ii) ANNs-driven FL scheme with leader election (AFwLE).

B. Simulation Results

1) Performance comparisons between SNN and ANN: For evaluating the performance of the proposed framework, we first compare the accuracy between FL-based on SNN and FL-based on ANN. Note that we build the ANN model that has similar framework with the SNN model.

Figs. 2(a) and (b) compare the accuracy between SNN and ANN on Fashion-MNIST and MNIST, respectively. From Fig. 2(a), we can observe that SFLwE-based on SNNs shows a progressive improvement in accuracy from 73.38% to 90.28%, whereas AFLwE-based on ANNs exhibits a rise from 69.64% to 86.86% over 40 training epochs. Over the course of 40 epochs, SFLwE exhibits an overall accuracy improvement of 16.9% (from 73.38% to 90.28%), while AFLwE shows a total increase of 17.22% (from 69.64% to 86.86%). The similar trend can be found in MINIST dataset, as shown in Fig. 2(b). SFLwE exhibits a substantial total accuracy increase of 50.33% throughout the training, moving from 47.77% to 98.1%. Conversely, AFLwE demonstrates a total improvement of 39.79%, from 57.73% to 97.52%.

2) Performance comparisons between SFLwE and SFL: Fig. 2(c) and Fig. 2(d) verify the advantages of the leader election. We compare the performance of the proposed SFLwE with SFLE (i.e., without leader election) by recording the time required to achieve 90.28% accuracy on the Fashion-MNIST dataset and 98.1% accuracy on the MNIST dataset, respectively. Given the same training time, incorporating leader selection into FL models enhances time efficiency across various accuracy benchmarks. We observe that the proposed SFLwE achieves 90.28% accuracy in 1120 seconds, while SFLE (i.e., without leader election) requires about 1461 seconds to reach the same level of accuracy. This represents a 23.34% reduction in time consumption with the SFLwE scheme.

3) Comparisons of energy consumption between SNNs and ANNs: We evaluate the energy consumption of model training by the cumulative Floating Point Operations (FLOPS). This effectively mirrors the count of multiply-and-accumulate (MAC) or dot product tasks conducted for every inference on each input [16]. Note that the energy consumption is illustrated in Section II-B considers the energy consumption of hardware,

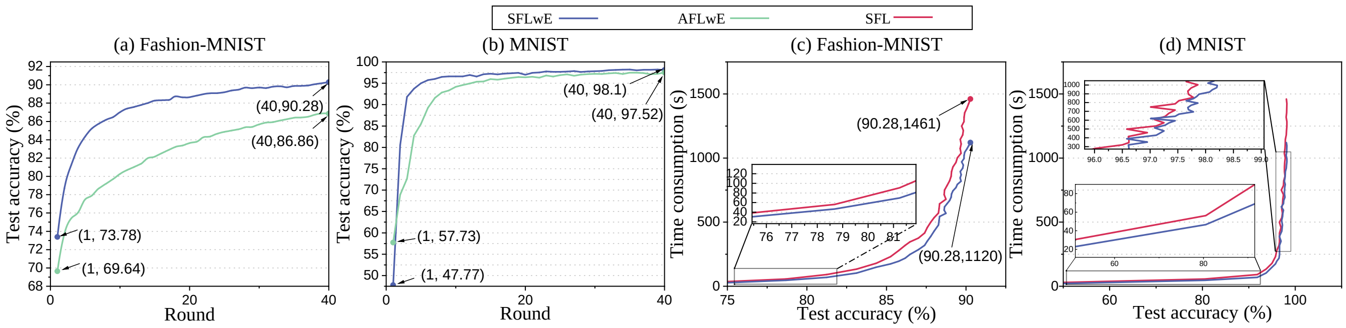

Fig. 2: Performance on various methods and datasets. (a) and (b): global accuracy vs. training round with SNN and ANN on Fashion-MNIST and MNIST datasets, respectively. (c) and (d): Time consumption vs. global accuracy on Fashion-MNIST and MNIST datasets, respectively.

and this section only focuses on the energy consumption of neural networks (i.e., convolution and fully connected layers in SNNs and ANNs). Specifically, the calculations of SNNs are based on binary (i.e., $\mathcal{S}$=0 or $\mathcal{S}$=1), the MAC operations can be replaced by the accumulate (AC) operations.

TABLE I: The energy consumption of single sample inference on ANN and SNN.

Method	Energy (μj)
SNN	**0.042** (1.0×)
ANN	2.763 (65.8×)

For a convolution layer in an ANNs or SNNs that uses a kernel size of $k \times k$, with N input channels, M output channels, and an input feature map size of $I \times I$, the total number of FLOPS in a convolution is given by $F_{con} = I^2 \times N \times k^2 \times M$. In the case of a fully connected layer with N inputs and M outputs, the total number of operations is calculated as $F_{fc} = N \times M$. The energy consumption of ANNs and SNNs can be expressed as $E_{ANN} = (F_{con} + F_{fc}) \times E_{MAC}$ and $E_{SNN} = F_{con} \times E_{MAC} + (F_{con} \times R_{con} + F_{fc} \times R_{fc}) \times T \times E_{AC}$, where $E_{MAC} = 3.2$ pJ and $E_{AC} = 0.1$ pJ are the energy consumption of MAC and AC, respectively, and more details can be found in [16]. R_{con} and R_{fc} denote the firing rate of spike in convolution and fully connected layer, respectively. The average firing rates of spikes were determined by performing inference on the last sample of the test set, resulting in firing rates of 0.157 ($Conv2d$), 0.044 ($Conv2d$), 0.08 (FC), and 0.09 (FC), respectively. As shown in TABLE I, the ANN inference on one sample consumes about 65.8 times energy than the SNN.

V. Conclusion

In this work, we have proposed an SNN-driven decentralized FL framework deployed on a UAV swarm, aimed at minimizing energy and time consumption while training a high-performance FL model. Specifically, we have first conducted a detailed analysis of how the SNN-driven FL can reduce the energy consumption and its challenges (i.e., non-differentiability) when deployed in UAV swarm. We have then presented an approximate derivative algorithm to solve this challenge. Furthermore, we have designed a leader election scheme to mitigate the negative effects of wireless factors and accelerate the aggregation of the FL model. Simulation results have shown that our proposed scheme can outperform baseline schemes in terms of model performance, energy and time consumption.

References

[1] S. Javed, A. Hassan, R. Ahmad, W. Ahmed, R. Ahmed, A. Saadat, and M. Guizani, "State-of-the-art and future research challenges in uav swarms," *IEEE Internet of Things Journal*, pp. 1–1, 2024.

[2] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622–1658, 2021.

[3] P. Li, H. Zhang, Y. Wu, L. Qian, R. Yu, D. Niyato, and X. Shen, "Filling the missing: Exploring generative ai for enhanced federated learning over heterogeneous mobile edge devices," *IEEE Transactions on Mobile Computing*, 2024.

[4] B. Jiang, J. Du, G. Yang, C. Jiang, C. Liu, and Y. Tian, "Energy-efficient dynamic device scheduling for over-the-air federated learning in uav swarms," in *IEEE Global Communications Conference*, 2023, pp. 01–06.

[5] T. Zeng, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated learning in the sky: Joint power allocation and scheduling with uav swarms," in *IEEE International Conference on Communications*, 2020, pp. 1–6.

[6] Y. Shen, Y. Qu, C. Dong, F. Zhou, and Q. Wu, "Joint training and resource allocation optimization for federated learning in uav swarm," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2272–2284, 2023.

[7] C.-H. Nguyen, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, V.-D. Nguyen, Y. Xiao, and E. Dutkiewicz, "Encrypted data caching and learning framework for robust federated learning-based mobile edge computing," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2024.

[8] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2021.

[9] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE International Conference on Computer Communications*, 2019, pp. 1387–1395.

[10] 3GPP, "Overall description of Radio Access Network (RAN) aspects for Vehicle-to-everything (V2X) based on LTE and NR," 3rd Generation Partnership Project (3GPP), Technical Specification (TS), July 2020, version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3601

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[12] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, "Incorporating learnable membrane time constant to enhance learning of spiking neural networks," in *2021 IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2641–2651.

[13] K.-R. Koch and K.-R. Koch, "Bayes' theorem," *Bayesian Inference with Geodetic Applications*, pp. 4–8, 1990.

[14] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[15] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[16] P. Panda, S. A. Aketi, and K. Roy, "Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization," *Frontiers in Neuroscience*, vol. 14, p. 535502, 2020.