Adaptive Model Pruning for Communication and Computation Efficient Wireless Federated Learning

Zhixiong Chen, *Graduate Student Member, IEEE*, Wenqiang Yi, *Member, IEEE*,
Hyundong Shin, *Fellow, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

***Abstract*—Most existing wireless federated learning (FL) studies focused on homogeneous model settings where devices train identical local models. In this setting, the devices with poor communication and computation capabilities may delay the global model update and degrade the performance of FL. Moreover, in the homogenous model settings, the scale of the global model is restricted by the device with the lowest capability. To tackle these challenges, this work proposes an adaptive model pruning-based FL (AMP-FL) framework, where the edge server dynamically generates sub-models by pruning the global model for devices' local training to adapt their heterogeneous computation capabilities and time-varying channel conditions. Since the involvement of diverse structures of devices' sub-models in the global model updating may negatively affect the training convergence, we propose compensating for the gradients of pruned model regions by devices' historical gradients. We then introduce an age of information (AoI) metric to characterize the staleness of local gradients and theoretically analyze the convergence behaviour of AMP-FL. The convergence bound suggests scheduling devices with large AoI of gradients and pruning the model regions with small AoI for devices to improve the learning performance. Inspired by this, we define a new objective function, i.e., the average AoI of local gradients, to transform the inexplicit global loss minimization problem into a tractable one for device scheduling, model pruning, and resource block (RB) allocation design. Through detailed analysis, we derive the optimal model pruning strategy and transform the RB allocation problem into equivalent linear programming that can be effectively solved. Experimental results demonstrate the effectiveness and superiority of the proposed approaches. The proposed AMP-FL is capable of achieving 1.9x and 1.6x speed up for FL on MNIST and CIFAR-10 datasets in comparison with the FL schemes with homogeneous model settings.**

***Index Terms*—Device scheduling, federated learning, model pruning, resource management**

I. Introduction

The explosive growth of data generated at edge devices motivated deploying advanced machine-learning techniques in future wireless networks to exploit the data for serving diverse applications, e.g., autonomous driving and the metaverse [2]. Federated learning (FL) is a promising distributed learning framework that enables multiple edge devices to learn a shared global model collaboratively without exposing their private data [3]. However, implementing FL in practical wireless networks encounter two main challenges: 1) *Scarce Wireless Resources*: The limited wireless resource only allows a tiny proportion of devices to be scheduled in each learning round [4]. Since the local data distributions among devices are generally heterogeneous, partial device participation may lead to biased model aggregation and degrade the learning performance of FL [5]. 2) *Heterogenous Devices*: In practice, edge devices are drastically different in computation and communication capabilities. Most existing wireless FL studies focused on homogeneous model settings where all devices train identical models in each round. In this setting, the devices with poor capabilities delay global aggregation and slow down the learning convergence, as well as restrict the scale of the global model due to their resource bottlenecks [6]. It is worth mentioning that although the personalized FL approaches [6], [7] were developed to enable devices to train heterogeneous local models, they aim to train a customized local model for each device based on their individual local data distribution that may not generalize well on the classes out of their local data classes. When a device predicts classes are not in its local data, the personalized model shows lower performance than the generalized global model. Thus, this work mainly focuses on training a generalized shared global model while mitigating the straggler effect in the homogeneous local model setting. To address these problems, wireless FL needs a heterogeneous local model setting that is able to adapt devices' computation and communication capabilities.

A. Related Works

To tackle the limited wireless resources problem, existing works have developed various solutions to reduce bandwidth consumption, e.g., device scheduling [8]–[10], model compression [11]–[14], and knowledge distillation [15], [16]. The device scheduling approach selects part of the devices to engage in the learning process of each round to alleviate the communication burden. Specifically, the joint device scheduling and resource management approaches in [8] and [9] effectively reduced the energy consumption and convergence time of wireless FL, respectively. In [10], the co-design of device selection and wireless networks significantly

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC), U.K., under Grant EP/W004100/1 and Grant EP/W034786/1. Part of this work was presented at the IEEE Global Communications Conference (GLOBECOM), 2023 [1]. (Corresponding author: Arumugam Nallanathan)

Zhixiong Chen and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. Arumugam Nallanathan is also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggido 17104, Korea. (emails: {zhixiong.chen, a.nallanathan}@qmul.ac.uk).

W. Yi is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (email: wy23627@essex.ac.uk).

Hyundong Shin is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, Yongin-si, Gyeonggido 17104, Republic of Korea (e-mail: hshin@khu.ac.kr).

improved the learning accuracy of wireless FL. While device scheduling can effectively alleviate communication pressure for wireless FL, uploading the entire model parameters still poses a challenge for devices with poor channel conditions. To address this issue, the model compression approach reduces the uploaded data size for devices by quantizing devices' parameters with fewer bits or only uploading partial parameter elements to the server. In [11], the co-design of gradient compression at devices and reconstruction at the edge server significantly reduced the communication overhead and obtained a satisfactory learning performance for FL. The gradient sparsification scheme in [12] compressed the resultant sparse gradient to a low-dimensional vector to reduce the bandwidth consumption. In [13], the stochastic quantization approach significantly alleviated the communication burden and improved the convergence performance for FL. The model quantization approach in [14] achieved a tradeoff between learning accuracy and communication time by adjusting devices' quantization levels proportional to their communication conditions. While model compression effectively relieves the communication pressure for devices, they introduce additional noise during model aggregation and ultimately degrades the final model's accuracy. Knowledge distillation (KD)-based FL enables collaboratively training by exchanging light knowledge distilled on a public dataset between devices and the edge server. Specifically, the adaptive mutual KD-based FL approach in [15] substantially decreased the communication overheads for FL and obtained similar performance as centralized learning. By mixing the local training data to generate a distillation dataset to empower the FL process, the KD approach in [16] significantly accelerates the learning speed. Nevertheless, the prerequisite of the public dataset may leave these KD-based FL approaches infeasible for many practical scenarios since a carefully engineered public dataset may not always be available. In addition, the above device scheduling, model compression, and knowledge distillation approaches do not reduce the model complexity, and the computation overhead is still high for devices.

To learn a generalized shared global model while allowing devices to train heterogeneous local models that adapted their communication and computation capabilities, model pruning-based FL approaches were developed to reduce the resource demands for devices and achieve an approximate performance of the original models. Existing model pruning works can be categorised into unstructured weight pruning [17]–[19] and structured model pruning [20]–[22]. Specifically, the weight pruning approach prunes the weight parameters in the fully connected (FC) layer of the deep neural network (NN) to achieve both parameters and computation load reduction. The weight importance-aware pruning method in [17] removed the unimportant weights in deep NN, which effectively reduced the model size incurring only a small performance loss. The random pruning mechanism in [18] significantly reduced device communication and computation overhead and avoided model overfitting. In [19], the pruning ratio and bandwidth allocation scheme improved the convergence speed of FL. However, these unstructured weight pruning approaches may be ineffective in reducing the computation load of the convolution NN since the pruned weight connections are from the FC layers. In contrast, the computation overhead is mainly concentrated in convolution layers. For instance, in VGG-16, the FC layers account for 90% of the total parameters but only occupy less than 1% of the overall floating point operations [23]. Moreover, the unstructured pruning approach usually results in irregular weight matrixes in the pruned models that are difficult to compress, which requires specialized hardware and software libraries to accelerate the training speed [24]. To effectively decrease computation and communication overhead, the structured model pruning approach [20]–[22], [25], [26] was developed to prune both filters in convolution layers and neurons in FC layers to generate sub-models for devices to train. Note that in centralized learning, pruning filters in convolution layers have been demonstrated can effectively accelerate the learning speed without sacrificing too much accuracy [24], [27]. The random sub-model generation scheme in [20] effectively decreased the server-to-client communication and device-side computation costs. The static model pruning approach in [21], [25] or local model composition approach in [28] distributed heterogeneous sub-models to devices for training and then aggregated them into a global inference model, which effectively reduced resource consumption for FL. The model shrinking and gradient compression approach in [26] enabled the local model training with elastic computation and communication overheads. The model pruning method in [22] dynamically adjusted the model size for resource-limited devices and significantly improved the cost-efficiency of FL. Although these structured model pruning approaches effectively reduced the communication and computation overhead for wireless FL, the different parts in the global model may not be trained evenly across devices. This may induce the different parts in the global model to drift toward different devices and degrade the learning performance of FL.

B. Motivations and Contributions

Although the approaches in [8]–[16] with homogeneous model settings effectively alleviate the communication bottlenecks for wireless FL, the computation overhead is still high for devices. This may restrict devices with poor computation capabilities from engaging in the training process and thus degrades the learning performance. In addition, the model pruning approaches [17]–[22] do not consider the model and data heterogeneity in the model pruning process, and the global model's parameters are not evenly trained across devices. Consequently, different regions in the global model are biased toward different devices. This may degrade the final model's accuracy, especially under high data heterogeneity. To tackle these issues, this work jointly designs the wireless network and learning mechanism to enhance the learning performance for FL. Specifically, we propose an adaptive model pruning approach to dynamically generate sub-models for devices' local training by pruning the global model to adapt devices' heterogeneous computation capabilities and time-varying channel conditions in the learning process. To mitigate the negative effect of diverse structures of the sub-models from affecting the learning convergence, we propose

compensating the gradients of devices' pruned model regions by their historical gradients to improve learning performance. The main contributions of this paper are listed as:

- We propose an adaptive model pruning-based FL (AMP-FL) framework, which dynamically prunes the global model to generate sub-models for adapting devices' communication and computation capabilities in the learning process. This framework effectively reduces communication and computation overhead for devices at the same time, enabling efficient FL over heterogeneous devices. To prevent the diverse sub-model structures from affecting the learning convergence, we propose compensating for the gradients of pruned regions by devices' historical gradients. In addition, we theoretically analyze the relationship between the pruning ratio and communication & computation load.
- We define an age of information (AoI) metric to characterize local gradients' staleness and theoretically analyze AMP-FL's convergence bound. Differing from existing convergence analysis works with full device participation, e.g., [22], [26], our convergence bound is based on the more practical partial device participation situation and characterizes how the AoI of devices' gradients affect the learning performance of FL. The bound indicates that scheduling devices with large AoI and pruning the global model regions with small AoI are able to improve learning performance. Based on this, we define a new objective function, i.e., the average square of AoI of devices' gradients, and transform the inexplicit global loss minimization problem into a tractable one for guiding device scheduling, model pruning, and resource block allocation design. Note that the proposed approach aims to minimize the average AoI of devices' gradients and achieves better learning performance than existing works in [17], [19] that reduce the model pruning ratio.
- To solve the transformed problem, we first find the optimal model pruning policies for devices under a given RB allocation policy. On this basis, we transform it into an equivalent linear programming problem that can be effectively solved with polynomial time complexity. In addition, to improve the implementation feasibility of AMP-FL in practical wireless networks, we propose a memory-friendly AMP-FL equivalent to AMP-FL but with a low memory size requirement of the edge server.
- We conduct extensive simulations on two real-world datasets, i.e., MNIST and CIFAR-10, to verify the effectiveness of AMP-FL. Specifically, compared to the FL algorithms with the homogeneous local model settings, the proposed AMP-FL is able to provide 1.9x and 1.6x speed up on MNIST and CIFAR-10, respectively. The proposed model pruning and device scheduling approach also obtains higher learning accuracy and faster convergence speed than the benchmark schemes.

C. Organizations

The rest of this paper is organized as follows: Section II introduces the system model, the proposed AMP-FL framework, and the problem formulation. In Section III, we present the convergence analysis results and the problem transformation. Section IV illustrates the proposed model pruning, device scheduling, and RB allocation algorithm. Section V evaluates the effectiveness of the proposed approaches by simulations. The conclusion is presented in Section VI.

II. System Model and Learning Mechanism

This work considers a typical wireless FL system, as shown in Fig. 1, where K devices are orchestrated by an edge server to collaboratively train a shared global machine learning model, $\boldsymbol{w}$, by periodically uploading local gradient information to the edge server for global model update instead of transmitting the raw training data. To mitigate the negative effect of stragglers on learning performance, this work allows devices to train heterogeneous local models adapted to their computation and communication capabilities. The local models are obtained by pruning the global model using the proposed structured model pruning strategy (in Section IV-A) that dynamically adjusts the local models during the learning process with respect to devices' individual heterogeneous computation capabilities and time-varying communication conditions.

We assume that the global model can be partitioned into I disjoint regions indexed by $\mathcal{I} = \{1, 2, \cdots, I\}$, where each model region i is either one filter in convolution layers or one neuron in the fully-connected layers. Let $\boldsymbol{w}^{(i)}$ $(i \in \mathcal{I})$ denote the i-th region of the global model. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. Each device k $(k \in \mathcal{K})$ has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples. The entire dataset is denoted by $\mathcal{D} = \cup_{k=1}^{K} \mathcal{D}_k$ with $D = \sum_{k=1}^{K} D_k$ data samples. For any data sample $\zeta = (\boldsymbol{x}, y) \in \mathcal{D}$, a loss function $f(\boldsymbol{x}, y; \boldsymbol{w})$ is utilized to capture the fitting performance of model $\boldsymbol{w}$ on the input-output data pair $(\boldsymbol{x}, y)$. Thus, the local loss function of device k $(k \in \mathcal{K})$, i.e., $F_k(\boldsymbol{w})$, is given by $F_k(\boldsymbol{w}) = \frac{1}{D_k} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_k} f(\boldsymbol{x}, y; \boldsymbol{w})$. The global loss function is given by $F(\boldsymbol{w}) = \sum_{k=1}^{K} a_k F_k(\boldsymbol{w})$, where a_k is the weight of device k such that $a_k \geq 0$ and $\sum_{k=1}^{K} a_k = 1$. Similar to many existing works, e.g., [5], [17], [29], we consider a balanced size of local datasets by setting $a_k = \frac{1}{K}$, $\forall k \in \mathcal{K}$. The goal of the FL system is to train a shared global model $\boldsymbol{w}$ so as to minimize the global loss $F(\boldsymbol{w})$ on the whole dataset $\mathcal{D}$, i.e., $\min_{\boldsymbol{w}} F(\boldsymbol{w})$.

A. Federated Learning with Adaptive Model Pruning

To improve the communication and computation efficiency for wireless FL, this work proposes a novel AMP-FL framework to adaptively generate sub-models for devices to train, as shown in Fig. 1. In addition, to alleviate the adverse effects of diverse structures of local models and partial participation in the learning performance, we propose compensating the gradients of pruned model regions and unscheduled devices by devices' historical gradients. The effectiveness of this gradient compensation mechanism is evaluated in Section V. To this end, the edge server maintains a gradient array $\{\boldsymbol{G}_{k,t} : \forall k \in \mathcal{K}\}$ that caches the latest received gradients from devices. The learning process consists of T global rounds and running the following five steps in each round t $(t \in \{0, 1, \cdots, T-1\})$:

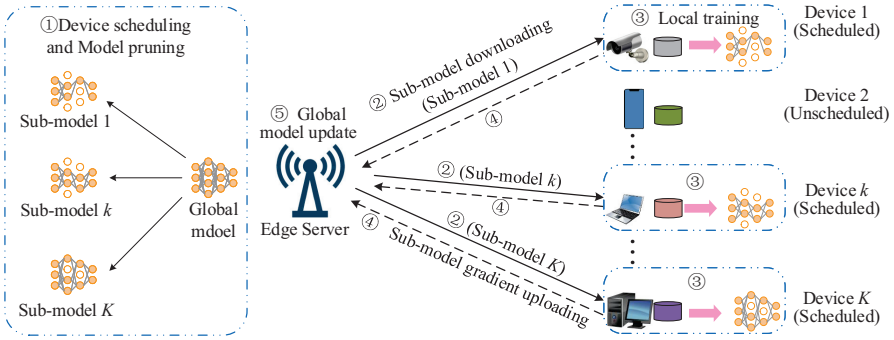


Fig. 1. Illustration of the considered wireless FL system with adaptive model pruning.

1) **Device Selection and Model Pruning**: The edge server selects a subset of devices to engage in the current round. Denote $\alpha_{k,t} \in \{0,1\}$ as the selection indicator of device k in t-th round, where $\alpha_{k,t}=1$ represents device k is selected, $\alpha_{k,t}=0$ otherwise. For ease of presentation, let $\boldsymbol{S}_t = \{k : \alpha_{k,t}=1, \forall k \in \mathcal{K}\}$ denote the device scheduling decision in round t. After device selection, the edge server prunes the global model to generate sub-models for the scheduled devices according to their computing and communication capabilities. Let $\boldsymbol{m}_{k,t} = \{m_{k,t}^{(i)} : i=1,2,\cdots,I\}$ denote the pruning mask of device k in round t, where $m_{k,t}^{(i)}=1$ represents that the i-th region of the global model is preserved in device k's sub-model, $m_{k,t}^{(i)}=0$ otherwise. Thus, the sub-model of device k can be denoted as $\boldsymbol{w}_{k,t} = \boldsymbol{w}_t \odot \boldsymbol{m}_{k,t}$, where $\odot$ denote the element-wise product.

2) **Local Model Downloading**: Each selected device downloads its sub-model from the edge server.

3) **Local Model Training**: Each selected device trains its sub-model by performing λ-steps stochastic gradient descent (SGD). Specifically, device k ($k \in \boldsymbol{S}_t$) updates the i-th region ($\forall i \in \mathcal{I}, m_{k,t}^{(i)}=1$) of its model as

$$\boldsymbol{w}_{k,t,l+1}^{(i)} = \boldsymbol{w}_{k,t,l}^{(i)} - \eta \nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}), l \in \{0,1,\cdots,\lambda-1\}, \tag{1}$$

where $\boldsymbol{w}_{k,t,l}^{(i)}$ is the i-th region of device k's local model in the l-th iteration in round t with $\boldsymbol{w}_{k,t,0}^{(i)} = \boldsymbol{w}_t^{(i)}$, and η is the learning rate. In (1), the stochastic gradient $\nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)})$ is given by $\nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}) = \frac{1}{L_b}\sum_{\zeta \in \mathcal{B}_{k,t,l}} \nabla f(\boldsymbol{w}_{k,t,l}^{(i)}, \zeta)$, where $\mathcal{B}_{k,t,l}$ is a mini-batch data uniformly sampled from $\mathcal{D}_k$ with $L_b = |\mathcal{B}_{k,t,l}|$ data samples.

4) **Local Gradient Uploading**: After finishing local training, each scheduled device k ($k \in \boldsymbol{S}_t$) uploads its cumulative local gradient, i.e., $\tilde{\boldsymbol{g}}_{k,t} = \{\tilde{\boldsymbol{g}}_{k,t}^{(i)} : \forall i \in \mathcal{I}, m_{k,t}^{(i)} = 1\}$, to the edge server, where $\tilde{\boldsymbol{g}}_{k,t}^{(i)}$ is given by $\tilde{\boldsymbol{g}}_{k,t}^{(i)} = \sum_{l=0}^{\lambda-1} \nabla \tilde{F}_k(\boldsymbol{w}_{k,t,l}^{(i)}) = \frac{1}{\eta}(\boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{k,t,\lambda}^{(i)})$.

5) **Global Model Update**: After receiving the local gradients from the scheduled devices, the edge server updates the gradient array as follows:

$$\boldsymbol{G}_{k,t}^{(i)} = \begin{cases} \tilde{\boldsymbol{g}}_{k,t}^{(i)}, & \alpha_{k,t} m_{k,t}^{(i)} = 1, \\ \boldsymbol{G}_{k,t-1}^{(i)}, & \alpha_{k,t} m_{k,t}^{(i)} = 0, \end{cases} \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{2}$$

Then, the edge server updates the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t}^{(i)}, \forall i \in \mathcal{I}$.

B. Communication and Computation Load Model

Let $\mathcal{C}$ represent the number of float-point operations (FLOPs) required to process one data sample on the global model, and $\mathcal{Q}$ indicate the number of global model parameters. In each round, devices train heterogeneous local models generated by pruning the global model to adapt to their communication and computation capabilities. For any device k ($k \in \mathcal{K}$) in round t with pruning mask $\boldsymbol{m}_{k,t}$, its pruning ratio is given by

$$\beta_{k,t} = 1 - \frac{1}{I}\sum_{i=1}^{I} m_{k,t}^{(i)}. \tag{3}$$

In fact, (3) indicates the ratio of pruned filters and neurons in the global model. To avoid introducing layer-wise hyperparameters, the proposed AMP-FL uses the same pruning ratio for every convolution or FC layer. Given the pruning ratio, AMP-FL removes a corresponding ratio of filters and neurons in each convolutional layer and FC layer to generate sub-models. In the following, we analyze the number of parameters and FLOPs for device k's sub-model from the perspective of convolution and FC layers.

1) For the l-th convolution layer in the global model with C_l filters, the number of parameters in this layer is $\mathcal{Q}_{g,l} = (K_c^2 \times C_{l-1} + 1) \times C_l$ which contains $K_c^2 \times C_{l-1} \times C_l$ weight parameters and C_l bias parameters; the number of FLOPs is $\mathcal{C}_{g,l} = 2K_c^2 HW C_{l-1} \times C_l$, where C_{l-1} is the number of filters in the $(l-1)$-th layer, K_c is the filter width (assumed to be symmetric), H and W are the height and width of the input feature maps [30]. For device k's sub-model with pruning ratio $\beta_{k,t}$, the number of parameters contained in its sub-model in this layer is $\mathcal{Q}_{k,l} = (1-\beta_{k,t})^2 K_c^2 \times C_{l-1} \times C_l + (1-\beta_{k,t}) \times C_l \approx (1-\beta_{k,t})^2 \mathcal{Q}_{g,l}$, and the number of FLOPs is $\mathcal{C}_{k,l} = 2(1-\beta_{k,t})^2 C_{l-1} C_l K_c^2 WH = (1-\beta_{k,t})^2 \mathcal{C}_{g,l}$.

2) For the l-th FC layer in the global model with N_l neurons, the number of parameters is $\mathcal{Q}_{g,l} = (N_{l-1}+1) \times N_l$ which contains $N_{l-1} \times N_l$ weight parameters and N_l bias parameters; the number of FLOPs is $\mathcal{C}_{g,l} = 2N_{l-1} \times N_l$, where N_{l-1} is the number of neurons in the $(l-1)$-th FC layer. For device k's sub-model with pruning ratio $\beta_{k,t}$, the number of parameters contained in its sub-model is $\mathcal{Q}_{k,l} = (1-\beta_{k,t})^2 \times$

$N_{l-1} \times N_l + (1-\beta_{k,t}) \times N_l \approx (1-\beta_{k,t})^2 \mathcal{Q}_{g,l}$, and the number of FLOPs is $\mathcal{C}_{k,l} = 2(1-\beta_{k,t})^2 \times N_{l-1} \times N_l = (1-\beta_{k,t})^2 \mathcal{C}_{g,l}$.

Note that in the above analysis, we approximate the number of parameters of both convolution and FC layers in the sub-model to be the ratio, i.e., $(1-\beta_{k,t})^2$, of that in the original global model. This is because the number of bias parameters is far less than that of weight parameters. According to the above analysis, for each device k with pruning ratio β_k, the number of parameters and FLOPs for its sub-model can be approximately scaled by $(1-\beta_{k,t})^2$ of the global model. That is, the number of parameters of device k's sub-model is

$$\mathcal{Q}_k = (1-\beta_{k,t})^2 \mathcal{Q}, \tag{4}$$

and the corresponding number of FLOPs required to process one data sample is

$$\mathcal{C}_k = (1-\beta_{k,t})^2 \mathcal{C}. \tag{5}$$

C. Learning Latency Model

In the following, we characterize the per-round learning latency model for the proposed AMP-FL, including computation and communication latency.

1) **Computation Latency**: We consider the CPU adopted to perform local training on each device. Let f_k be the CPU frequency of device k. Each CPU cycle can process n_k FLOPs. Thus, the computation time[1] of device k is

$$\mathcal{T}^{\mathrm{L}}_{k,t} = \frac{\lambda L_b \mathcal{C}_k}{f_k n_k} = \frac{\lambda L_b (1-\beta_{k,t})^2 \mathcal{C}}{f_k n_k}. \tag{6}$$

2) **Communication Latency**: This work considers the orthogonal frequency division multiple access is utilized with R resource blocks (RBs) for devices to transmit their gradient information. The RBs are indexed by $\mathcal{R} = \{1, 2, \cdots, R\}$. Let $\boldsymbol{z}_{k,t} = (z^{(1)}_{k,t}, z^{(2)}_{k,t}, \cdots, z^{(R)}_{k,t})$ denote the RB allocation decision of device k in round t, where $z^{(r)}_{k,t} \in \{0,1\}$, $z^{(r)}_{k,t} = 1$ represents that the r-th RB is allocated to device k, $z^{(r)}_{k,t} = 0$ otherwise. For ease of representation, we use $\boldsymbol{Z}_t = (\boldsymbol{z}_{1,t}, \boldsymbol{z}_{2,t}, \cdots, \boldsymbol{z}_{K,t})$ denote the RB allocation decisions for all devices in round t. Denote p_k as the transmit power of device k. Let $h_{k,t}$ represent the channel gain between device k and the edge server, and it remains unchangeable within one round but varies independently over rounds. Thus, the transmit rate of device k is $r_{k,t}(\boldsymbol{z}_{k,t}) = \sum_{r=1}^{R} z^{(r)}_{k,t} B \log_2 \left(1 + \frac{p_k h_{k,t}}{I_r + B N_0}\right)$, where B is the bandwidth of each RB, N_0 is the noise power spectral density. I_r is the interference caused by devices located in other service areas not participating in the FL process and using the same resource block [9], [10]. We consider that each device can only occupy at most one RB, and each RB can be accessed by at most one device. Thus, $\sum_{r=1}^{R} z^{(r)}_{k,t} \leq 1$ and $\sum_{k=1}^{K} z^{(r)}_{k,t} \leq 1$. Each parameter in devices' local gradients is quantized by q bits. Thus, the transmit time of device k to upload its gradient information is

$$\mathcal{T}^{\mathrm{U}}_{k,t} = \frac{\mathcal{Q}_k q}{r_{k,t}(\boldsymbol{z}_{k,t})} = \frac{(1-\beta_{k,t})^2 \mathcal{Q} q}{r_{k,t}(\boldsymbol{z}_{k,t})}. \tag{7}$$

Note that the above analysis ignored the model pruning, global model updating, and sub-model downloading latencies since the edge server is usually computationally powerful and has more transmit power than devices [13], [17]. The model pruning, global model updating, and sub-model downloading latencies are negligible compared to the above communication and computation latency. In addition, it is worth mentioning that the proposed algorithm in Section IV can be directly generalized in the case with non-negligible sub-model download latency by simply adding the sub-model download latency into the time constraint (8a).

D. Problem Formulation

This work focuses on improving the performance of the proposed AMP-FL by minimizing the global loss value after T global training rounds, i.e., $\mathbb{E}[F(\boldsymbol{w}_T)]$, where $\boldsymbol{w}_T$ is the global model in round T. Specifically, we jointly optimize the device scheduling, model pruning, and RB allocation strategies under latency and wireless resource restrictions. The optimizing problem is given by

$$\mathcal{P}: \quad \min_{\{\boldsymbol{S}_t, \boldsymbol{Z}_t, \boldsymbol{m}_t\}_{t=0}^{T-1}} \ \mathbb{E}[F(\boldsymbol{w}_T)] \tag{8}$$

$$\text{s. t.} \quad \mathcal{T}^{\mathrm{L}}_{k,t} + \mathcal{T}^{\mathrm{U}}_{k,t} \leq \mathcal{T}_{\max}, \forall k \in \mathcal{K}, \forall t, \tag{8a}$$

$$\sum\nolimits_{r=1}^{R} z^{(r)}_{k,t} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{8b}$$

$$\sum\nolimits_{k=1}^{K} z^{(r)}_{k,t} \leq 1, \forall t, \tag{8c}$$

$$z^{(r)}_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{8d}$$

$$\beta_{k,t} = 1 - \frac{1}{I} \sum\nolimits_{i=1}^{I} m^{(i)}_{k,t}, \forall k \in \mathcal{K}, \forall t, \tag{8e}$$

$$0 \leq \beta_{k,t} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{8f}$$

$$m^{(i)}_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \forall t, \tag{8g}$$

$$\alpha_{k,t} \in \{0,1\}, \forall k \in \mathcal{K}, \tag{8h}$$

where (8a) stipulates that the per-round latency cannot surpass its maximum allowed delay, $\mathcal{T}_{\max}$. (8b), (8c), and (8d) impose restrictions on the RB allocation decisions. (8e) characterizes the relationship between pruning policy and model pruning ratio for devices. (8f) prevents the model pruning ratio from exceeding 1 or lessening 0 since the edge server can prune at most the entire model or not prune for the global model. (8g) and (8h) correspond to the constraints related to the model pruning and device scheduling indicator domains, respectively.

Problem $\mathcal{P}$ is a typical integer programming that involves multi-dimensional discrete variables and is intractable to solve. In addition, solving problem $\mathcal{P}$ requires an explicit form of $\mathbb{E}[F(\boldsymbol{w}_T)]$ with respect to the device selection ($\boldsymbol{S}_t$), model pruning ($\boldsymbol{m}_t$), and RB allocation ($\boldsymbol{Z}_t$) policies, which is almost impossible since the evolution of the model vector is extremely complex during the training process. To this end, similar to many existing works, e.g., [6], [8]–[10], [29], we turn to find

[1]It is worth mentioning that the relationship between pruning ratio and computation time in (6) is not evident in the GPU-based model training. In practice, devices may adopt GPU or the hybrid of CPU and GPU for local training, and how to accurately characterize the computation time in this case is a promising future research direction.

an upper bound of the global loss function and optimize it for global loss minimization in Section III.

III. Convergence Analysis and Problem Transformation

In this section, we theoretically characterize the convergence behaviour of AMP-FL to explore how the device schedule, model pruning, and RB allocation policies affect its learning performance. Based on the obtained convergence bound, we define a new objective function, i.e., the AoI for local gradients, to transform problem $\mathcal{P}$ into a tractable one for guiding the device selection, model pruning, and RB allocation design.

A. Convergence Analysis

This subsection analyzes the convergence behaviour of AMP-FL. For ease of analysis, we define $\nabla F_k(\boldsymbol{w}_{k,t,l}) = \frac{1}{D_k}\sum_{(\boldsymbol{x},y)\in\mathcal{D}_k}\nabla f(\boldsymbol{x},y;\boldsymbol{w}_{k,t,l})$ as the full gradient of device k in the l-th iteration of round t. Let $F(\boldsymbol{w}^*)$ be the loss function of the optimal global model $\boldsymbol{w}^*$. Note that we use the latest received gradients of the pruned model regions of scheduled devices and unscheduled devices to update the global model. To characterize the impact of the staleness of devices' gradients on the learning performance, we define an AoI metric to identify the staleness of devices' gradients. Specifically, the AoI of the gradient in the i-th region of device k is denoted by $\tau_{k,t}^{(i)}$, which evolves as

$$\tau_{k,t}^{(i)} = \begin{cases} \tau_{k,t-1}^{(i)}+1, & \alpha_{k,t}m_{k,t}^{(i)}=0, \\ 0, & \alpha_{k,t}m_{k,t}^{(i)}=1, \end{cases} \forall k\in\mathcal{K},\forall i\in\mathcal{I}. \quad (9)$$

To facilitate the analysis, we make the following standard assumptions which are widely used in the existing FL literature, e.g., [8]–[10], [31].

Assumption 1. All the local loss functions, $F_k(\boldsymbol{w})$ ($\forall k\in\mathcal{K}$), are L-smooth. That is, for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $\|\nabla F_k(\boldsymbol{w})-\nabla F_k(\boldsymbol{v})\|\le L\|\boldsymbol{w}-\boldsymbol{v}\|$.

Assumption 2. All the local loss functions, $F_k(\boldsymbol{w})$ ($\forall k\in\mathcal{K}$), are μ-strongly convex. That is, for all $\boldsymbol{v}$ and $\boldsymbol{w}$, $F_k(\boldsymbol{v})\ge F_k(\boldsymbol{w})+\langle\nabla F_k(\boldsymbol{w}),\boldsymbol{v}-\boldsymbol{w}\rangle+\frac{\mu}{2}\|\boldsymbol{v}-\boldsymbol{w}\|^2$.

Assumption 3. For the mini-batch data $\mathcal{B}_{k,t}$ that uniformly sampled from $\mathcal{D}_k$ on device k ($k\in\mathcal{K}$), the resulting stochastic gradient $\tilde{\nabla}F_k(\boldsymbol{w}_t)$ is an unbiased estimation of the full gradient $\nabla F_k(\boldsymbol{w}_t)$, i.e., $\mathbb{E}[\tilde{\nabla}F_k(\boldsymbol{w}_t)]=\nabla F_k(\boldsymbol{w}_t)$, and its variance is bounded by σ^2, i.e., $\mathbb{E}\|\tilde{\nabla}F_k(\boldsymbol{w}_t)-\nabla F_k(\boldsymbol{w}_t)\|^2\le\sigma^2$.

Assumption 4. The expected squared norm of devices' gradients is uniformly bounded by G^2, i.e., $\|\nabla F_k(\boldsymbol{w}_t)\|^2\le G^2$, for all $k=1,2,\cdots,K$ and $t=0,1,\cdots,T-1$.

Before illustrating the convergence results of the proposed AMP-FL, we introduce two lemmas in the following to assist our convergence analysis.

Lemma 1. *Let Assumption* 1, 3, *and* 4 *hold, and the learning rate satisfies* $\eta\le\frac{1}{2\lambda L}$, *the averaged drift of the local models from the global model after* l *iterations is bounded as*

$$\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\sum_{i=1}^{I}\mathbb{E}\|\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)}}^{(i)}-\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)}\|^2 \le 4(\lambda-1)I(2\eta^2\lambda G^2+\eta^2\sigma^2). \quad (10)$$

Proof. Please see Appendix A. □

Lemma 2. *Let Assumption* 1, 3, *and* 4 *hold, the averaged difference between the global model parameters in two different rounds is bounded as*

$$\sum_{k=1}^{K}\sum_{i=1}^{I}\mathbb{E}\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}_{t-\tau_{k,t}^{(i)}}^{(i)}\|^2 \le 3\eta^2\Big((\lambda^2+(\lambda-1)I)\sigma^2 + (\lambda^2+2\lambda(\lambda-1)I)G^2\Big)\sum_{k=1}^{K}\sum_{i=1}^{I}(\tau_{k,t}^{(i)})^2. \quad (11)$$

Proof. Please see Appendix B. □

Based on the above two lemmas, the one-round convergence bound of AMP-FL is derived as:

Theorem 1. *Let Assumption* 1, 3, *and* 4 *hold, and the learning rate satisfies* $\eta\le\frac{1}{2\lambda L}$, *the one-round convergence bound is*

$$\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)] \le (-\frac{1}{2}\eta+L\eta^2\lambda)\lambda\|\nabla F(\boldsymbol{w}_t)\|^2+c_1 + \frac{15\eta^2Lc_2}{4K}\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2, \quad (12)$$

where $c_1=4\eta(\lambda-1)IG^2+\big(4\eta^2L(\lambda-1)I+\frac{3}{4}\eta\lambda\big)\sigma^2$, $c_2=\lambda^2(\sigma^2+G^2)+(\lambda-1)I(\sigma^2+2\lambda G^2)$.

Proof. Please see Appendix C. □

According to Theorem 1, the summation of the square of each region's AoI in the local gradients, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$, is a crucial factor that negatively affects the one-round convergence bound of AMP-FL. Minimizing $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$ through carefully designing the device scheduling and model pruning strategies is capable of narrowing the convergence bound for improving the learning performance. We have the following remark for the device scheduling and model pruning design.

Remark 1. In practical wireless networks, only a small proportion of devices can be scheduled in each round due to the limited bandwidth resources. For device scheduling, one should schedule the devices that have a large summation of AoI over their model regions, i.e., $\sum_{i=1}^{I}(\tau_{k,t-1}^{(i)}+1)^2$, since these devices are the main contributors for the term of $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$. In addition, for a scheduled device, one should preserve the model regions with large AoI while pruning the regions with small AoI.

Based on Theorem 1, we further analyze the convergence bound of AMP-FL after T-rounds as follows:

Corollary 1. *Let Assumption* 1-4 *hold, the* T*-rounds convergence bound of AMP-FL is*

$$\mathbb{E}[F(\boldsymbol{w}_T)-F(\boldsymbol{w}^*)] \le (1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^T\mathbb{E}[F(\boldsymbol{w}_0)-F(\boldsymbol{w}^*)] + \frac{1-(1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^T}{\eta\lambda\mu-2L\eta^2\lambda^2\mu}c_1 + \frac{15}{4}\eta^2Lc_2\sum_{t=0}^{T-1}(1-\eta\lambda\mu+2L\eta^2\lambda^2\mu)^{T-1-t}\times \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2. \quad (13)$$

Proof. Please see Appendix D. □

From Corollary 1, the expected gap between $F(\boldsymbol{w}_T)$ and the optimal loss $F(\boldsymbol{w}^*)$ is bounded by three terms: 1) The initial gap between the global loss and the optimal loss. 2) A constant term related to the system hyperparameters caused by multiple local iterations ($\lambda > 1$) and stochastic gradient error. 3) The cumulative AoI of local gradients over T training rounds. The last term is highly related to model pruning, device scheduling, and wireless resource allocation policies. To minimize the global loss function, one can minimize the last term on the right-hand side (RHS) of (13) through jointly designing the model pruning, device scheduling, and wireless resource allocation strategies. However, directly minimizing this term is impractical because it requires obtaining devices' channel state information during the entire learning course at the start of FL. To minimize the global loss, we have:

Remark 2. Similar to many existing works, e.g., [8]–[10], the available wireless resource and devices are independent across different rounds in problem $\mathcal{P}$. Based on Theorem 1 and Corollary 1, we provide a reasonable objective function by decoupling the long-term problem into the training round level, i.e., $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$, which directly minimizes the upper bound on $\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]$ and achieves global loss minimization.

B. Problem Transformation

According to the convergence analysis results in Remark 1 and Remark 2, we transform problem $\mathcal{P}$ into minimize $\sum_{k=1}^{K}\sum_{i=1}^{I}(1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2$ in each round (which is equivalent to maximize $\sum_{k=1}^{K}\sum_{i=1}^{I}\alpha_{k,t}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2$) for device scheduling, model pruning, and RB allocation policies design. Since $\alpha_{k,t}=\sum_{r=1}^{R}z_{k,t}^{(r)}\in\{0,1\}$, we have $\sum_{k=1}^{K}\sum_{i=1}^{I}\alpha_{k,t}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2=\sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{r=1}^{R}z_{k,t}^{(r)}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2$. In other words, when the RB allocation policy is determined, the device scheduling policy can be directly computed by $\alpha_{k,t}=\sum_{r=1}^{R}z_{k,t}^{(r)}$. Therefore, we transform problem $\mathcal{P}$ into the following problem:

$$\widetilde{\mathcal{P}}:\ \max_{\boldsymbol{Z}_t,\boldsymbol{m}_t}\ \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{r=1}^{R}z_{k,t}^{(r)}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2 \tag{14}$$
$$\text{s. t.}\ \ (8a),(8b),(8c),(8d),(8e),(8f),(8g).$$

Problem $\widetilde{\mathcal{P}}$ is a typical integer programming that is challenging to solve. In the following section, we develop an effective algorithm with polynomial time complexity to address its optimal solution. Note that problem $\widetilde{\mathcal{P}}$ is to maximize the overall AoI across different devices and model regions. According to the evolution of AoI in Eq. (9), the model regions or devices are less frequently updated in the previous rounds have large AoI and thus tend to be selected to be updated in the current round. Thus, problem $\widetilde{\mathcal{P}}$ helps regulate the updating frequency of diverse regions across devices, making each model region evenly trained on different devices and improving the learning performance.

IV. Efficient Online Model Pruning and Resource Allocation

In this section, we develop an effective model pruning and RB allocation algorithm that solves problem $\widetilde{\mathcal{P}}$. To this end, we first derive the optimal model pruning policy for devices under any given RB allocation policy. Based on the optimal pruning policy, we transform problem $\widetilde{\mathcal{P}}$ into an equivalent linear programming problem which can be effectively solved. After that, to improve the implementation feasibility of AMP-FL in practical wireless networks, we propose a memory-friendly AMP-FL that is equivalent to the proposed AMP-FL in Section II-A but with a low memory requirement of the edge server.

A. Optimal Model Pruning Policy

For any given RB allocation policy $\boldsymbol{Z}_t$, the model pruning policies of devices do not affect each other and independently contribute to the objective function. That is, the model pruning policy of each device can be solely optimized. Motivated by this, we decompose the model pruning optimization problem for each scheduled device k ($k\in\boldsymbol{S}_t$) from problem $\widetilde{\mathcal{P}}$ as follows:

$$\mathcal{P}_1:\ \max_{\boldsymbol{m}_{k,t}}\ \sum\nolimits_{r=1}^{R}z_{k,t}^{(r)}\sum\nolimits_{i=1}^{I}m_{k,t}^{(i)}(\tau_{k,t-1}^{(i)}+1)^2 \tag{15}$$
$$\text{s. t.}\ \ (8e),(8f),(8g),$$
$$\frac{\lambda L_b(1-\beta_{k,t})^2\mathcal{C}}{f_k n_k}+\frac{(1-\beta_{k,t})^2\mathcal{Q}q}{r_{k,t}(\boldsymbol{z}_{k,t})}\le\mathcal{T}_{\max}, \tag{15a}$$

where constraint (15a) is obtained by rewrite constraint (8a). Problem $\mathcal{P}_1$ is a typical unweighted knapsack problem. Based on constraint (8e) and (15a), the pruning policy of device k should satisfy $\frac{1}{I}\sum_{i=1}^{I}m_{k,t}^{(i)}\le\sqrt{\mathcal{T}_{\max}/(\frac{\lambda L_b\mathcal{C}}{f_k n_k}+\frac{\mathcal{Q}q}{r_{k,t}(\boldsymbol{z}_{k,t})})}$. Moreover, based on constraint (8f) and (8g), the number of preserved model regions, i.e., $\sum_{i=1}^{I}m_{k,t}^{(i)}$, should be an integer and not exceed total number regions of the global model, i.e., I. According to (15), one should preserve model regions as much as possible to increase the objective function value. Thus, the optimal pruning policy of device k satisfy

$$\frac{1}{I}\sum_{i=1}^{I}m_{k,t}^{(i)}=\min\left(\left\lfloor\sqrt{\frac{\mathcal{T}_{\max}}{\frac{\lambda L_b\mathcal{C}}{f_k n_k}+\frac{\mathcal{Q}q}{r_{k,t}(\boldsymbol{z}_{k,t})}}}\right\rfloor,1\right), \tag{16}$$

where $\lfloor\cdot\rfloor$ is the floor function which outputs the largest integer that does not exceed its input. From (16), when the RB allocation policy is given, the number of preserved regions for device k's sub-model is fixed. For the optimal model pruning policy of device k, we have the following remark:

Remark 3. The optimal pruning policy for device k ($k\in\boldsymbol{S}_t$) is to preserve the model regions with large AoI while pruning the model regions with small AoI for maximizing the objective function of problem $\mathcal{P}_1$.

Note that, similar to many existing works, e.g., [20]–[22], this work adopts the width scaling approach to prune the global model, which removes a certain number of filters or neurons in each convolution layer or FC layer to generate a sub-model. To avoid introducing layer-wise hyperparameters, we use the

Algorithm 1 Adaptive Model Pruning Algorithm

1: **Initialization:** The AoI of device k's gradients in all model regions, i.e., $\{\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}\}$. The RB allocation policy of device k, $\boldsymbol{z}_{k,t}$.
2: Solve the optimal number of preserved model regions $\bar{\beta}_{k,t} = \sum_{i=1}^{I} m_{k,t}^{(i)}$ based on (16).
3: **for** each layer in the global model **do**
4: Sort the regions in this layer according to their AoI (i.e., $\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}$) in an descending order and then preserve the first $\bar{\beta}_{k,t}$ model regions and prune other regions.

same pruning ratio for every convolution or fully-connected layer. For each convolution layer or FC layer, we sort the filters or neurons based on their AoI in descending way, then gradually select the corresponding ratio (computed as (16)) of regions and remove the remaining regions. Let $\mathcal{L}$ denote the set of layers in the global model. Here, for each layer. many sorting algorithms can be utilized, e.g., Quicksort and Introsort, with a meagre time complexity of $\mathcal{O}(I_l \log I_l)$, where I_l is the number of filters or neurons in l-th layer of the global model. Due to $\sum_{l\in\mathcal{L}} I_l \log I_l \leq \sum_{l\in\mathcal{L}} I_l \log(\max_{l\in\mathcal{L}} I_l) = I \log(\max_{l\in\mathcal{L}} I_l)$, the model pruning process has a meagre time complexity of $\mathcal{O}(I \log(\max_{l\in\mathcal{L}} I_l))$. We summarize the detailed steps of model pruning in Algorithm 1.

B. Optimal Resource Block Allocation

According to the above analysis, the optimal pruning strategy for each device k ($k \in \mathcal{K}$) can be solved when it accesses any RB r ($r \in \mathcal{R}$) using Algorithm 1, denoted as $\boldsymbol{m}_{k,t,r}^{*} = \{m_{k,t,r}^{(i,*)} : \forall i \in \mathcal{I}\}$. Based on this, we compute the optimal model pruning policies for all devices when they access any RB (i.e., $\{\boldsymbol{m}_{k,t,r}^{*} : \forall r \in \mathcal{R}, \forall k \in \mathcal{K}\}$) and then substitute them into problem $\widetilde{\mathcal{P}}$. Thus, $\widetilde{\mathcal{P}}$ can be simplified as the following equivalent RB allocation problem:

$$\mathcal{P}_2: \quad \max_{\boldsymbol{Z}_t} \sum_{k=1}^{K}\sum_{r=1}^{R} z_{k,t}^{(r)} \sum_{i=1}^{I} m_{k,t,r}^{(i,*)} (\tau_{k,t-1}^{(i)} + 1)^2 \tag{17}$$
$$\text{s. t.} \quad (8b), (8c), (8d).$$

Problem $\mathcal{P}_2$ is a typical integer programming which is difficult to solve. Below we reformulate it as a maximum weight bipartite matching problem and find its optimal solution. To this end, we construct a complete and balanced bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{K} \cup \bar{\mathcal{R}}$ is the vertex set, and $\mathcal{E}$ is the set of edges that connect the vertices in K and $\bar{\mathcal{R}}$. In graph $\mathcal{G}$, each vertex k ($k \in \mathcal{K}$) corresponds to a device k. $\bar{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}_v$ is an extended set of $\mathcal{R}$, where each vertex r ($r \in \mathcal{R}$) corresponds to r-th RB, $\mathcal{R}_v$ is the virtual vertex set used to construct a balanced bipartite graph. The weight of edges is given by

$$\Omega_{k,r} = \begin{cases} \sum_{i=1}^{I} m_{k,t,r}^{(i,*)} (\tau_{k,t-1}^{(i)} + 1)^2, & \text{if} \quad k \in \mathcal{K}, r \in \mathcal{R}, \\ 0, & \text{else.} \end{cases} \tag{18}$$

Based on the above defined bipartite graph $\mathcal{G}$, problem $\mathcal{P}_2$ can be transformed to find a maximum weight perfect matching of graph $\mathcal{G}$. Let $\theta_{k,r} \in \{0, 1\}$ be the edge connecting vertex k and vertex r, where $\theta_{k,r} = 1$ denotes RB r is assigned to device k, and $\theta_{k,r} = 0$ otherwise. Denote $\boldsymbol{\theta}_k = \{\theta_{k,1}, \theta_{k,2}, \cdots, \theta_{k,R}\}$ by the edge connection indicator of device k to all RBs. The bipartite matching problem is given by:

$$\mathcal{P}_3: \quad \max_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_K} \sum_{k=1}^{K} \sum_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r} \Omega_{k,r} \tag{19}$$
$$\text{s. t.} \quad \sum_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r} = 1, \tag{19a}$$
$$\sum_{k=1}^{K} \theta_{k,r} = 1, \tag{19b}$$
$$\theta_{k,r} \in \{0, 1\}, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}. \tag{19c}$$

Note that any solution of problem $\mathcal{P}_3$ corresponds to a perfect matching of graph $\mathcal{G}$. The constraints (19a), (19b), and (19c) are corresponding to the constraints (8b), (8c), and (8d), respectively. To find the optimal solution of problem $\mathcal{P}_3$, an intuitive approach is to calculate the objective value of all perfect matching of graph $\mathcal{G}$, and let the matching with maximum objective value as the final RB allocation policy. However, this approach may be infeasible in practice since there is a total of $K!$ perfect matching of graph $\mathcal{G}$, which has an exponential time complexity since $K! > \sqrt{2\pi K}(\frac{K}{e})^K$. By relaxing the integrality constraint (19c), problem $\mathcal{P}_3$ can be relaxed as the following linear programming:

$$\mathcal{P}_4: \quad \max_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_K} \sum_{k=1}^{K} \sum_{r=1}^{|\bar{\mathcal{R}}|} \theta_{k,r} \Omega_{k,r} \tag{20}$$
$$\text{s. t.} \quad (19a), (19b),$$
$$0 \leq \theta_{k,r} \leq 1, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}. \tag{20a}$$

It is worth mentioning that in problem $\mathcal{P}_4$, each row in the coefficient matrix corresponding to (19a) and (19b) only contains a '1'. This implements that each square submatrix of this coefficient has determinant equal to 0, 1, or -1. Thus, this coefficient matrix is a totally unimodular matrix. Based on [32], the optimal solution of problem $\mathcal{P}_4$ is an integer solution. That is, the optimal solution of $\mathcal{P}_4$ equals to the optimal solution of problem $\mathcal{P}_3$. Therefore, we directly solve problem $\mathcal{P}_4$ to obtain the optimal solution of $\mathcal{P}_3$. Since problem $\mathcal{P}_4$ is a linear programming, we use the current matrix multiplication time algorithm [33] to solve it with a time complexity of $\mathcal{O}((K^{2+1/6})^2)$.

C. Complexity Analysis and Implementation

In the above analysis, we first transform problem $\widetilde{\mathcal{P}}$ into an equivalent maximum weight perfect bipartite matching problem, i.e., problem $\mathcal{P}_3$. Then, we further transform problem $\mathcal{P}_3$ into its equivalent linear programming $\mathcal{P}_4$. It is worth mentioning that these are two equivalent transformations and do not change the optimality of problem $\widetilde{\mathcal{P}}$. Thus, the optimal solution of problem $\widetilde{\mathcal{P}}$ can be addressed by first solving the optimal solution of problem $\mathcal{P}_4$. When the optimal solution of problem $\mathcal{P}_4$ is found, the optimal RB allocation is determined. Furthermore, the optimal device scheduling policy can be computed by $\alpha_{k,t}^{*} = \sum_{r=1}^{R} z_{k,t}^{(r),*}$ ($\forall k \in \mathcal{K}$), and the optimal model pruning policy of each device can be determined by Algorithm 1. For clarity, we summarize the detailed steps for solving problem $\widetilde{\mathcal{P}}$ in Algorithm 2. In Algorithm 2, constructing the linear programming problem $\mathcal{P}_4$ requires running $K \times R$ times of Algorithm 1 to calculate the optimal

Algorithm 2 Efficient Device Scheduling, Model Pruning, and RB Allocation Algorithm

Initialization: The AoI of devices' gradients in all model regions, $\{\tau_{k,t}^{(i)}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}\}$.
Solve the optimal model pruning policy for each device k ($k \in \mathcal{K}$) at each RB r ($r \in \mathcal{R}$) using Algorithm 1.
Construct the linear programming problem $\mathcal{P}_4$.
Solve $\mathcal{P}_4$ by the current matrix multiplication time algorithm [33] and obtain the optimal solution $\{\theta_{k,r}^*, \forall k \in \mathcal{K}, \forall r \in \bar{\mathcal{R}}\}$.
Compute the RB allocation policy for each device k ($k \in \mathcal{K}$) as $\boldsymbol{z}_{k,t}^* = \{z_{k,t}^{(r,*)}, \forall r \in \mathcal{R}\}$ where $z_{k,t}^{(r,*)} = \theta_{k,r}^*$.
Compute the device scheduling policy as $\boldsymbol{S}_t^* = \{\alpha_{k,t}^* = 1, \forall k \in \mathcal{K}\}$ where $\alpha_{k,t}^* = \sum_{r=1}^{R} z_{k,t}^{(r,*)}$.
Find the optimal model pruning policies for each scheduled device $k \in \boldsymbol{S}_t^*$, denoted as $\{\boldsymbol{m}_{k,t}^*, \forall k \in \boldsymbol{S}_t^*\}$
return The device scheduling policy $\boldsymbol{S}_t^*$, model pruning policy $\boldsymbol{m}_{k,t}^*$, and RB allocation policy $\boldsymbol{z}_{k,t}^*$.

model pruning policy for each device k ($k \in \mathcal{K}$) at each RB r ($r \in \mathcal{R}$). Thus the overall time complexity to solve the problem $\widetilde{\mathcal{P}}$ is $\mathcal{O}\left(KRI \log I + (K^{2+1/6})^2\right)$.

Algorithm 3 Memory-friendly AMP-FL

Initialization: The edge server initials its gradient array $\bar{\boldsymbol{G}}_{-1} = \boldsymbol{0}$ and the global model $\boldsymbol{w}_0$, each device k ($k \in \mathcal{K}$) initial their gradient array as $\boldsymbol{G}_{k,-1} = \boldsymbol{0}$
Server side:
for $t = 0, 1, \cdots, T-1$ **do**
  Determine the scheduled devices and generate a sub-model for each scheduled device through model pruning.
  if Receive the gradient information from the selected devices **then**
    Update the gradient array $\bar{\boldsymbol{G}}_t$ as $\bar{\boldsymbol{G}}_t^{(i)} = \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} (\tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)})$
    Update the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \bar{\boldsymbol{G}}_t^{(i)}$.
  else
    $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$
Device side:
if Device k is scheduled **then**
  Download its corresponding sub-model from the edge server;
  for $l = 0, 1, \cdots, \lambda - 1$ **do**
    Perform local training according to (1);
  Compute the cumulative stochastic gradient $\tilde{\boldsymbol{g}}_{k,t}^{(i)} = \frac{1}{\eta}(\boldsymbol{w}_t^{(i)} - \boldsymbol{w}_{k,t,\lambda}^{(i)})$
  Upload the $\tilde{\boldsymbol{g}}_{k,t} - \boldsymbol{G}_{k,t-1}$ to the edge server.
  Update the gradient array $\boldsymbol{G}_{k,t}$ according to (2).

In practical wireless networks, implementing the proposed AMP-FL in Section II-A requires the edge server to maintain the gradient information for all devices. Thus, the memory size requirement of the edge server scales with the model size and the number of devices. With the increase in device number, the memory space of the edge server may be exhausted and thus restrict the scale of the FL system and the global model. To tackle this issue, we distribute the memory requirement to devices for forming a memory-friendly AMP-FL which is equivalent to the proposed AMP-FL in Section II-A. As a result, the edge server only need to maintain a single gradient array, $\bar{\boldsymbol{G}}_t$, to cache the aggregated local gradient information, and each device maintains a gradient array $\boldsymbol{G}_{k,t}$, to cache its previous latest gradient. Then we replace step 4) and step 5) in Section II-A with the following steps:

- Replace step 4) in Section II-A with: After finishing the local training process, each scheduled device k ($k \in \boldsymbol{S}_t$) uploads the difference between its current and previous gradient, i.e., $\bar{\boldsymbol{G}}_{k,t}^{(i)} = \tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}$, to the edge server.
- Replace step 5) in Section II-A with: After receiving devices' gradient information, the edge server updates the maintained gradient according to $\bar{\boldsymbol{G}}_t^{(i)} = \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum_{k=1}^{K} \bar{\boldsymbol{G}}_{k,t}^{(i)}$. Then, the edge server updates the global model as $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \bar{\boldsymbol{G}}_t^{(i)}$.

By replacing step 4) and step 5) in Section II-A with the above two steps, the edge server distributes the memory requirement to the devices. We summarise the steps of implementing this memory-friendly AMP-FL in Algorithm 3.

In the following theorem, we prove the equivalence of Algorithm 3 and the proposed AMP-FL in Section II-A.

Theorem 2. *Algorithm 3 is equivalent to the proposed AMP-FL in Section II-A.*

Proof. We prove Theorem 2 by mathematical induction approach. Firstly, the maintained gradient array $\bar{\boldsymbol{G}}_t$ at the edge server satisfies:

$$\begin{aligned}\bar{\boldsymbol{G}}_t^{(i)} &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum\nolimits_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} \bar{\boldsymbol{G}}_{k,t}^{(i)} \\ &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum\nolimits_{k=1}^{K} \alpha_{k,t} m_{k,t}^{(i)} \left(\tilde{\boldsymbol{g}}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}\right) \\ &= \bar{\boldsymbol{G}}_{t-1}^{(i)} + \frac{1}{K}\sum\nolimits_{k=1}^{K} \left(\boldsymbol{G}_{k,t}^{(i)} - \boldsymbol{G}_{k,t-1}^{(i)}\right). \end{aligned} \tag{21}$$

Note that at the beginning of the learning process, the devices' gradient array $\boldsymbol{G}_{k,-1}$ and the server's gradient array $\boldsymbol{G}_{-1}$ are all initialized with $\boldsymbol{0}$. Thus, when $t = 0$, we have

$$\begin{aligned}\bar{\boldsymbol{G}}_0^{(i)} &= \bar{\boldsymbol{G}}_{-1}^{(i)} + \frac{1}{K}\sum\nolimits_{k=1}^{K} \left(\boldsymbol{G}_{k,0}^{(i)} - \boldsymbol{G}_{k,-1}^{(i)}\right) \\ &= \frac{1}{K}\sum\nolimits_{k=1}^{K} \boldsymbol{G}_{k,0}^{(i)}. \end{aligned} \tag{22}$$

When $t = 1$,

$$\begin{aligned}\bar{\boldsymbol{G}}_1^{(i)} &= \bar{\boldsymbol{G}}_0^{(i)} + \frac{1}{K}\sum\nolimits_{k=1}^{K} \left(\boldsymbol{G}_{k,1}^{(i)} - \boldsymbol{G}_{k,0}^{(i)}\right) \\ &= \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,1}^{(i)} + \bar{\boldsymbol{G}}_0^{(i)} - \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,0}^{(i)} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,1}^{(i)}. \end{aligned} \tag{23}$$

Similarly, for $t > 1$, $\bar{\boldsymbol{G}}_t^{(i)} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t}^{(i)}$. Thus, the updated global model through Algorithm 3 is $\boldsymbol{w}_{t+1}^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \bar{\boldsymbol{G}}_t^{(i)} = \boldsymbol{w}_t^{(i)} - \eta \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{G}_{k,t}^{(i)}$, which equals the updated global model by the proposed AMP-FL in Section II-A. Thus, Algorithm 3 is equivalent to the AMP-FL algorithm in Section II-A. □

TABLE I
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
K	100	R	10
B	1MHz	N_0	-174dBm
n_k ($\forall k \in \mathcal{K}$)	4	h_0	-30dBm
q	32bits	η	0.05
τ	8	L_b	64
Q(CNN)	36,758	C_k(CNN)	782,816
$\mathcal{T}_{\max}$ (CNN)	0.1s	Q(VGG-11)	9,287,434
C_k(VGG-11)	362,285,568	$\mathcal{T}_{\max}$ (VGG-11)	20s
p_k($\forall k \in \mathcal{K}$)	30dBm		

V. Simulation Results

In this section, simulations are conducted to evaluate the performance of the proposed AMP-FL algorithm and device scheduling approach. If not specified, the default system settings are given as Table I. We consider an edge server situated at the centre of a circular area with a radius of 500m serving K randomly distributed devices. The channel gain is modelled as $h_{k,t} = h_0 \rho_{k,t} d_k^{-2}$, where d_k is the distance from device k to the edge server, $\rho_{k,t} \sim \text{Exp}(1)$ is the Rayleigh fading channel gain [6], [34]. For each device, its CPU frequency is uniformly selected from $\{0.85, 1.12, 1.2, 1.3\}$GHz. Similar to [9], [10], we do not compute the exact value of the interference ($I_m, \forall r \in \mathcal{R}$) since we mainly focus on FL system instead of other service areas. The inter-cell interference at each RB r, i.e., I_r, is randomly selected from the range of $\left[10^2 BN_0, 10^5 BN_0\right]$. For each device, we set its transmit power to $p_k = 30$dBm, and its CPU frequency is uniformly selected from $\{0.85, 1.12, 1.2, 1.3\}$GHz, and each CPU cycle can process 4 FLOPs.

We evaluate the proposed approaches on two typical classification tasks using MNIST and CIFAR-10 datasets. For the MNIST dataset, we train a convolution neural network (CNN) with the following structure: two 5×5 convolution layers with 6 and 16 channels, respectively, and each of them is followed by a 2×2 max-pooling layer; a 128-neuron FC layer; and a 10-unit softmax output layer. For CIFAR-10, we train a VGG-11 model [23]. Note that the original VGG-11 has 1000 output units. To adapt VGG-11 to the CIFAR-10 dataset, we remove its last max-pooling layer, then replace its FC layers as the following structure: two FC layers with 512 and 128 neurons; a 10-unit softmax output layer. We utilize a typical non-IID data partitioning method for both the above datasets as follows: we sort all data samples according to the label, then divide them into $sK/10$ shards and assign each device with s shard. By this means, each device obtains at most s types of data in the dataset. If not specified, $s = 2$. For all the above-illustrated two models, cross-entropy is used as the loss function.

A. Comparison of Model Pruning Strategies

In this subsection, we evaluate the proposed model pruning approach by comparing it with the following approaches under different device schedule numbers and pruning ratios: 1) The proposed model pruning without gradient compensation (Proposed-wGC): The edge server utilizes the proposed model pruning approach (i.e., Algorithm 1) to generate sub-models. However, the server only uses the received sub-model gradients from devices for global model updating without compensating the pruned model regions' gradients. The gradients of the pruned model regions of devices are set to zero for aligning the model architecture. That is, all devices' gradients have the same structure as the global model. 2) Importance-aware model pruning: In each round, the edge server removes the less important filters and neurons in the global model to generate sub-models. The importance score of each filter in the convolution layers is computed as the kernel weights summation. The importance score of each neuron in the FC layer is calculated as its connected input weights summation [35]. 3) Random pruning [18]: In each round, the server randomly prunes the global model to generate sub-models for devices based on their pruning ratios.

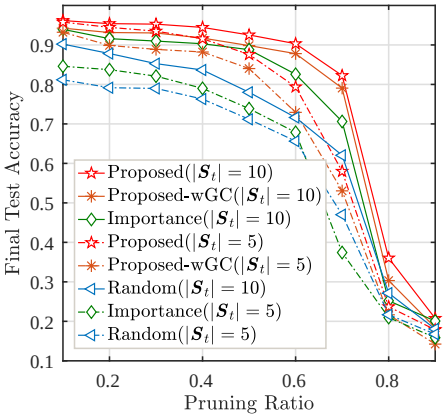


Fig. 3. Comparison of the final test accuracy of FL after 300 rounds of training with different pruning strategies on the MNIST dataset.

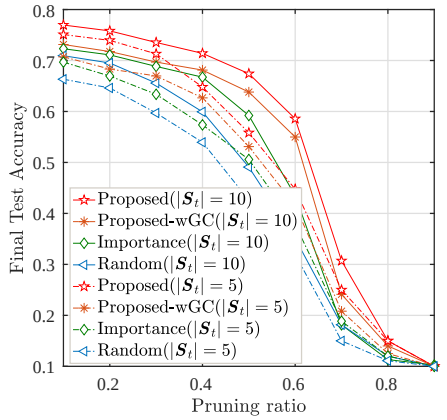


Fig. 5. Comparison of the final test accuracy of FL after 1000 rounds of training with different pruning strategies on the CIFAR-10 dataset.

Fig. 2 compares the learning performance of FL with different model pruning policies on the MNIST dataset. By setting the pruning ratio of all devices to 0.1, Fig. 2(a) and Fig. 2(b) test the performance of all the model pruning policies under $|\boldsymbol{S}_t| = 5$, Fig. 2(c) and Fig. 2(d) shown the learning performance under $|\boldsymbol{S}_t| = 10$. It is observed that the proposed approach achieves better learning performance than the benchmarks in terms of test accuracy and training loss. Compared to the benchmarks, the proposed approach improves over 10.9% and 3.3% accuracy when $|\boldsymbol{S}_t| = 5$ and $|\boldsymbol{S}_t| = 10$, respectively. In addition, the proposed pruning approach with gradient compensation performs better than that without gradient compensation. This demonstrates the effectiveness of the proposed gradient compensation mechanism. Fig. 3 shows how the pruning ratio affects the final accuracy of the global model trained under different pruning policies. Note that all the accuracy results in Fig. 3 are obtained by training the global model under corresponding pruning policies with 300 rounds.

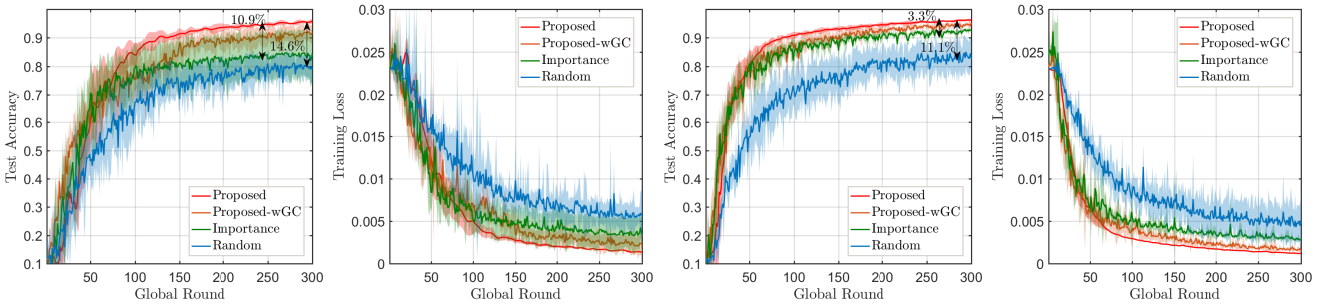


Fig. 2. Comparison of the learning performance of FL with different pruning strategies on the MNIST dataset: (a) Test accuracy, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, (b) Training loss, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, (c) Test accuracy, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$, (d) Training loss, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$.

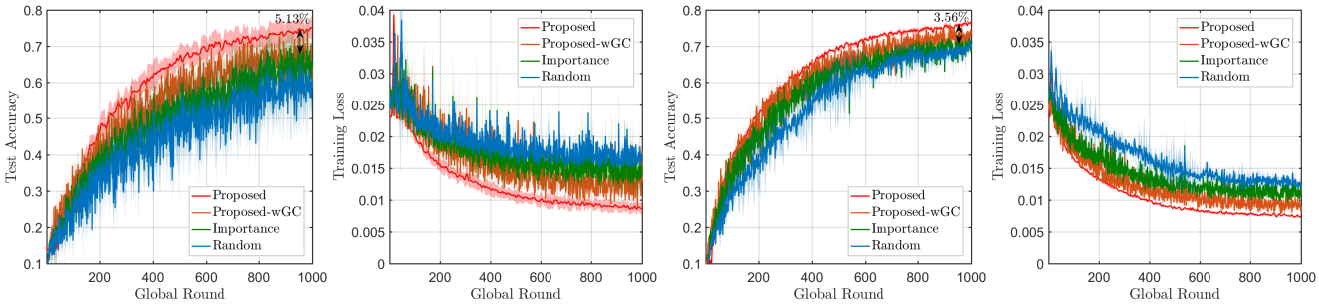


Fig. 4. Comparison of the learning performance of FL with different pruning strategies on the CIFAR-10 dataset: (a) Test accuracy, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, (b) Training loss, $|\boldsymbol{S}_t|= 5$, $\beta_{k,t} = 0.1$, (c) Test accuracy, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$, (d) Training loss, $|\boldsymbol{S}_t|= 10$, $\beta_{k,t} = 0.1$.

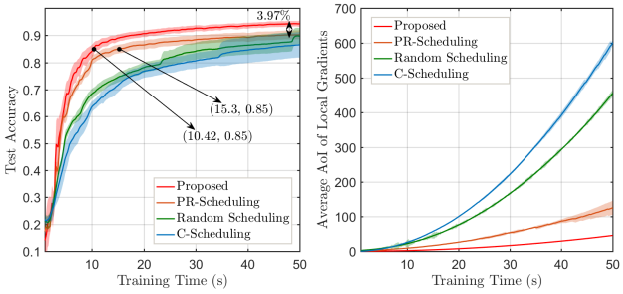
We can see that the proposed pruning approach outperforms the benchmarks under different pruning ratios and participant numbers. Moreover, for all pruning policies, the final accuracy under $|\boldsymbol{S}_t|= 10$ is higher than that under $|\boldsymbol{S}_t|= 5$. This indicates scheduling more devices in each round improves the learning performance of AMP-FL. In addition, the final model accuracy under all pruning policies decreases with the increase in the pruning ratio. This is because a larger pruning ratio induces that the sub-models have fewer parameters, and more filters and neurons have been trained fewer times. It is observed from Fig. 3 that a relatively small pruning ratio does not significantly hurt the global model accuracy while substantially reducing the communication costs. Under the given time, the pruned model can be trained for more rounds than the original model, resulting in better accuracy, which has been verified in Fig. 8 in Section V-C.

A similar comparison is conducted on the CIFAR-10 dataset in Fig. 4. It is observed the same conclusion as the results on MNIST, i.e., the proposed approach achieves better learning performance than the benchmarks in terms of test accuracy and training loss. When the pruning ratios of all approaches are set to be 0.1, the proposed approach is capable of boosting 5.13% and 3.56% accuracy under $|\boldsymbol{S}_t|= 5$ and $|\boldsymbol{S}_t|= 10$, respectively. It is worth mentioning that the proposed approach with gradient compensation outperforms that without gradient compensation. In addition, the proposed-wGC approach remains performs better than the other two benchmarks. This demonstrated the effectiveness of the proposed gradient compensation mechanism and model pruning approach. In addition, it is also observed that a small pruning ratio may not significantly hurt the global model accuracy. Thus, properly pruning the global model for devices allows more training rounds and achieves better learning performance than training the original model on devices, as shown in Fig. 9 in Section V-C.

The above results on MNIST and CIFAR-10 datasets demonstrate the effectiveness of the proposed adaptive model pruning approach and gradient compensation mechanism. For the practical implementation in wireless networks, these results suggest adaptively pruning the global model to enable each region of the model to be evenly trained across different devices, and utilizing the proposed gradient compensation mechanism helps enhance the learning performance of FL.

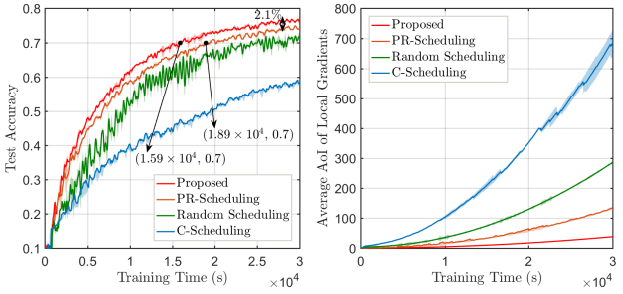
B. Comparison of Device Scheduling Policies

In this section, we evaluate the effectiveness of the proposed device scheduling and resource allocation approach by comparing it with: 1) Pruning ratio minimization-aware device scheduling (PR-scheduling) [17], [19]: In each round, the edge server selects a subset of devices that satisfies the latency constraint and has the minimal sum of the pruning ratio. 2) Channel gain-aware device scheduling (C-scheduling) [10]: The edge server schedules the devices with maximal channel gain and satisfies the latency constraint to perform training in each round. 3) Random scheduling. In each round, the



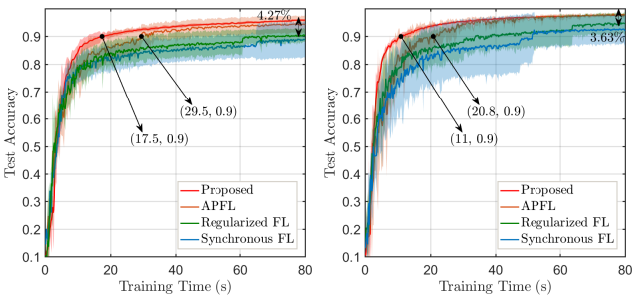
(a) (b)

Fig. 6. Comparison of learning performance for different device scheduling approaches on MNIST dataset.



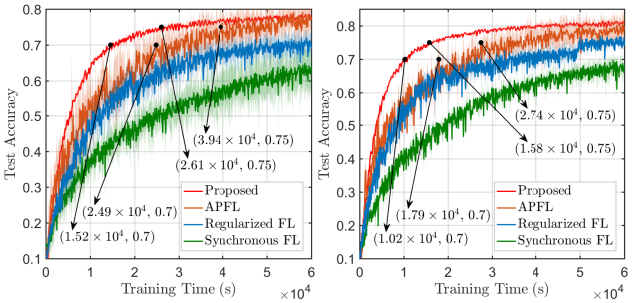
(a) (b)

Fig. 7. Comparison of learning performance for different device scheduling approaches on CIFAR-10 dataset.



(a) (b)

Fig. 8. Learning performance of different FL algorithms on the MNIST dataset: (a) $s = 2$, (b) $s = 3$.



(a) (b)

Fig. 9. Learning performance for different FL algorithms on the CIFAR-10 dataset: (a) $s = 2$, (b) $s = 3$.

edge server randomly selects a subset of devices and their corresponding RBs that satisfy the latency constraint.

Fig. 6 compares the proposed scheduling approach with the above three approaches on MNIST dataset. From Fig. 6(a), compared to the benchmarks, the proposed device scheduling achieves higher accuracy and faster convergence speed. Specifically, the proposed device scheduling approach boosts at least 3.97% accuracy than the benchmarks. Given the target accuracy is 85%, the proposed device scheduling approach only takes 10.42 seconds to achieve the target, while the best benchmark, i.e., the PR-scheduling scheme, requires 15.3 seconds. Compared to the benchmarks, the proposed approach is able to save 31.9% training time to obtain 85% test accuracy. The latent reason why the proposed device scheduling approach outperforms the benchmarks is illustrated in Fig. 6(b), which plots the average AoI of devices' local gradients. We find that the proposed method possesses the lowest average AoI of local gradients. In addition, for all the device scheduling algorithms, the one with lower AoI obtains higher learning accuracy. This phenomenon demonstrated the convergence results in Remark 1, which suggests minimizing the average AoI of local gradients to enhance the learning performance.

Fig. 7 evaluates the learning performance of all the device scheduling approaches on the CIFAR-10 dataset and shows the same conclusion as MNIST. From Fig. 7(a), the proposed approach achieves 2.1% accuracy improvement after 3×10^4 seconds of training. Given the target accuracy is 70%, the proposed device scheduling approach saves at least 15.87% training time compared to the benchmarks. In addition, Fig. 7(b) shows that the proposed device scheduling approach has the lowest average AoI of gradients compared to the benchmarks. This further demonstrated the correctness of the convergence results in Remark 1.

C. Overall Effectiveness

This subsection evaluates AMP-FL by comparing it to three FL algorithms as follows: 1) Synchronous FL [8]–[10]: The scheduled devices train the entire global model and upload the trained model to the edge server for aggregation. 2) Regularized FL [36]: Regularized FL utilizes a weight-based proximal term to limit the impact of local updates to tackle the data heterogeneity among devices. 3) Adaptive personalized FL (APFL) [37]: The selected devices train their local models and the received global model. After that, APFL integrates devices' local models and global model to create a personalized model for each device.

Fig. 8 shows the learning performance of AMP-FL and three benchmarks on MNIST dataset. Fig. 8(a) sets the data heterogeneity-related parameter to $s = 2$, i.e., each device in the system has at most two types of data samples of the MNIST dataset. We can see that AMP-FL significantly outperforms Synchronous FL and Regularized FL, i.e., it improves around 4.3% test accuracy compared to these two benchmarks. Although AMP-FL only obtains a slight accuracy improvement to the APFL approach, it converges fast than APFL. AMP-FL only takes 17.5s to achieve 90% accuracy,

while APFL takes 29.5s. That is, AMP-FL provides a 1.7x speed up compared to APFL. Fig. 8(b) compares the learning performance of all the FL algorithms under $s = 3$, drawing a similar conclusion to the setting of $s = 2$. Specifically, AMP-FL boosts 3.63% accuracy compared to Synchronous FL and Regularized FL and achieves a 1.9x speed up when the target accuracy is 90% compared to APFL. In addition, for all the FL algorithms, their learning performance under $s = 3$ is better than that under $s = 2$. This is because the high data heterogeneity would introduce higher variance in the global model update and degrade the learning performance.

Fig. 9 conducts a similar comparison on the CIFAR-10 dataset. From Fig. 9(a) with setting $s = 2$, when the target accuracy is 70% and 75%, the proposed AMP-FL is capable of providing a 1.6x and 1.5x speed up compared to the benchmarks, respectively. In Fig. 9(b), we set the data heterogeneity-related parameter to $s = 3$. It is observed that AMP-FL achieves a 1.75x and 1.7x speed up when the target accuracy is 70% and 75%, respectively. Moreover, we can see that the learning process of AMP-FL is more stable than that of the benchmarks since the shadow band of AMP-FL is slim than the benchmarks. The benefits come from the proposed gradient compensation mechanism and model pruning approach, which prevents the global model from being biased toward devices with high communication and computation capabilities. From the results in Fig. 8 and Fig. 9, dynamically adjusting the local models to adapt devices' computation and communication capabilities is an efficient approach to mitigate the straggler effects in practical wireless FL systems.

D. Impact of Wireless Resource on Learning Performance

In this subsection, we evaluate the impacts of the number of RBs on the learning performance of AMP-FL, including test accuracy and Average AoI of devices' gradients. Note that in this section, the results on MNIST and CIFAR-10 are achieved after 50 and 3×10^4 seconds of training, respectively.

In Fig. 10, we evaluate the effects of the number of RBs on the test accuracy and average AoI. From the results on MNIST dataset in Fig. 10(a), it is observed that the test accuracy of AMP-FL keeps increasing along with the increase in the number of RBs. This is because the increasing number of RBs allows more devices to participate in the learning process in each round. In addition, the average AoI of devices' gradients decreases with the increase in the number of RBs. According to the definition of AoI in (9), the AoI of a model region increases when the corresponding device is not selected or the model region is pruned. Increasing the number of resource blocks would increase the number of selected devices in each round, and thus more model regions are selected in each round. Consequently, the average AoI across devices would be reduced. The results on the CIFAR-10 dataset in Fig. 10(b) show a similar conclusion to Fig. 10(a), indicating that increasing the number of RBs helps improve test accuracy and reduce the average AoI of devices' gradients. These simulation results further verifies our theoretical analysis results in Remark 1, which suggests minimizing the average AoI of local gradients to enhance the learning performance.

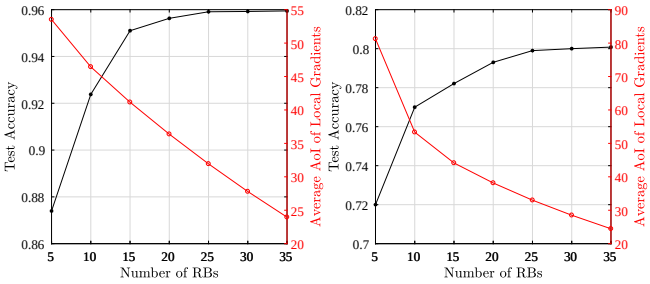


Fig. 10. Learning performance of the proposed AMP-FL under different number of RBs: (a) on MNIST dataset, (b) on CIFAR-10 dataset.

VI. Conclusion

In this work, we developed a novel AMP-FL framework which dynamically prunes the global model to generate sub-models adapted to devices' communication and computation capabilities. This framework is capable of simultaneously reducing communication and computation overhead for devices to enable efficient FL among heterogeneous devices. To prevent the diverse structures of pruned local models from affecting the training convergence, we proposed a gradient compensation mechanism to compensate for the gradients of pruned model regions by devices' historical gradients. We introduced an AoI metric to characterize the staleness of local gradients and analyzed the convergence bound of AMP-FL. The convergence bound suggests scheduling devices with large AoI and pruning the model regions with small AoI for devices in the per-round learning process. Based on this, we develop an effective device scheduling, model pruning, and RB allocation approach to enhance the learning performance of AMP-FL in wireless networks. Experimental results show that compared to the benchmark FL algorithms, the proposed AMP-FL is capable of achieving 1.9x and 1.6x speed up on MNIST and CIFAR-10 datasets, respectively.

Appendix

A. Proof of Lemma 1

If $\lambda = 1$, the inequality is trivially satisfied since $\boldsymbol{w}_{k,t,0}^{(i)} = \boldsymbol{w}_{k,t}^{(i)}$. For $\lambda > 1$, we have

$$\mathbb{E}\|\boldsymbol{w}_{k,t,l}^{(i)} - \boldsymbol{w}_t^{(i)}\|^2 = \mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \eta\nabla\tilde{F}_k(\boldsymbol{w}_{k,t,l-1}^{(i)}) - \boldsymbol{w}_t^{(i)}\right\|^2$$
$$\leq \mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)} - \eta\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\|^2 + \eta^2\sigma^2, \quad (24)$$

where the last inequality comes from adding and subtracting $\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})$ into $\nabla\tilde{F}_k(\boldsymbol{w}_{k,t,l-1}^{(i)})$, then using the triangle inequality and Assumption 3. Below we bound the first term in the above inequaltion as

$$\mathbb{E}\left\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)} - \eta\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\|^2$$
$$= \mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}\|^2 + \eta^2\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\|^2$$
$$- 2\mathbb{E}\left\langle\frac{1}{\sqrt{\lambda-1}}(\boldsymbol{w}_{k,t,l-1}^{(i)} - \boldsymbol{w}_t^{(i)}), \eta\sqrt{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\right\rangle$$

$$\begin{aligned}
&\overset{(a)}{\leq} (1+\frac{1}{\lambda-1})\mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2+\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})\|^2\\
&\leq (1+\frac{1}{\lambda-1})\mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2+2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\|^2\\
&+2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t,l-1}^{(i)})-\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\|^2\\
&\overset{(b)}{\leq} (1+\frac{1}{\lambda-1}+2\eta^2\lambda L^2)\mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2\\
&+2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\|^2\\
&\overset{(c)}{\leq}(1+\frac{3}{2(\lambda-1)})\mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2+2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\|^2,
\end{aligned} \tag{25}$$

where (a) comes from the triangle inequality, (b) comes from the L-smooth of loss functions, (c) is due to $\eta<\frac{1}{2\lambda L}$. Thus, we have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{w}_{k,t,l}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2&\leq (1+\frac{3}{2(\lambda-1)})\mathbb{E}\|\boldsymbol{w}_{k,t,l-1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2\\
&+2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_{k,t}^{(i)})\|^2+\eta^2\sigma^2.
\end{aligned} \tag{26}$$

By telescoping the above inequality, we have

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_{k,t,l}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2\\
&\leq \left(2\eta^2\lambda\mathbb{E}\|\nabla F_k(\boldsymbol{w}_t^{(i)})\|^2+\eta^2\sigma^2\right)\frac{(1+\frac{3}{2(\lambda-1)})^{\lambda-1}-1}{\frac{3}{2(\lambda-1)}}.
\end{aligned} \tag{27}$$

Since $(1+\frac{3}{2(\lambda-1)})^{\lambda-1}=(1+\frac{3}{2(\lambda-1)})^{\frac{2(\lambda-1)}{3}\frac{3}{2}}\leq e^{\frac{3}{2}}<5$ and $\frac{2(\lambda-1)}{3}<\lambda-1$, we have

$$\mathbb{E}\|\boldsymbol{w}_{k,t,l}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2\leq 4(\lambda-1)(2\eta^2\lambda G^2+\eta^2\sigma^2). \tag{28}$$

By substituting the above inequality into the left-hand-side of (10), the proof is completed.

B. Proof of Lemma 2

For $t_1>t_2$, let $\tilde{\eta}=\eta\lambda$, we have

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{w}_{t_1}^{(i)}-\boldsymbol{w}_{t_2}^{(i)}\|^2=\mathbb{E}\Big\|\sum_{t=t_2}^{t_1-1}(\boldsymbol{w}_{t+1}^{(i)}-\boldsymbol{w}_t^{(i)})\Big\|^2\\
&=\tilde{\eta}^2\mathbb{E}\Big\|\sum_{t=t_2}^{t_1-1}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\|^2\\
&\overset{(a)}{\leq} 3\tilde{\eta}^2(t_1-t_2)\sum_{t=t_2}^{t_1-1}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\\
&\qquad\qquad-\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\|^2\\
&+3\tilde{\eta}^2(t_1-t_2)\sum_{t=t_2}^{t_1-1}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\\
&\qquad\qquad-\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)}}^{(i)})\Big\|^2\\
&+3\tilde{\eta}^2(t_1-t_2)\sum_{t=t_2}^{t_1-1}\mathbb{E}\|\nabla F(\boldsymbol{w}_{t-\tau_{k,t}^{(i)}}^{(i)})\|^2\\
&\overset{(b)}{\leq} 3\tilde{\eta}^2(t_1-t_2)\sum_{t=t_2}^{t_1-1}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}L^2\mathbb{E}\Big\|\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)}-\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)}}^{(i)}\Big\|^2\\
&+3\tilde{\eta}^2(t_1-t_2)^2(\sigma^2+G^2),
\end{aligned} \tag{29}$$

where (a) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})$ and $\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)}}^{(i)})$ into $\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})$, (b) is due to Assumption 3 and 4. Based on Lemma 1, by substituting (10) into the above inequality, we have

$$\begin{aligned}
\mathbb{E}\Big\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}_{t-\tau_{k,t}^{(i)}}^{(i)}\Big\|^2&\leq 3\eta^2(\tau_{k,t}^{(i)})^2\Big((\lambda^2+(\lambda-1)I)\sigma^2\\
&+(\lambda^2+2\lambda(\lambda-1)I)G^2\Big).
\end{aligned} \tag{30}$$

Substituting the above inequation into the left-hand-side of (11), the proof is completed.

C. Proof of Theorem 1

Using the L-smooth property of local loss function, we have $\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]\leq\mathbb{E}\langle\nabla F(\boldsymbol{w}_t),\ \boldsymbol{w}_{t+1}-\boldsymbol{w}_t\rangle+\frac{L}{2}\mathbb{E}\left\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\right\|^2$. It is worth mentioning that both the inner product and norm can be broken down and reformulated as the sum of inner products and norms over all parameter regions, respectively. Thus, we have

$$\begin{aligned}
&\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right]\leq\sum_{i=1}^{I}\mathbb{E}\left\langle\nabla F(\boldsymbol{w}_t^{(i)}),\boldsymbol{w}_{t+1}^{(i)}-\boldsymbol{w}_t^{(i)}\right\rangle\\
&+\frac{L}{2}\sum\nolimits_{i=1}^{I}\mathbb{E}\|\boldsymbol{w}_{t+1}^{(i)}-\boldsymbol{w}_t^{(i)}\|^2\\
&=\underbrace{-\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}_t^{(i)}),\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\rangle}_{A_1}\\
&+\underbrace{\frac{L}{2}\tilde{\eta}^2\sum\nolimits_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum\nolimits_{k=1}^{K}\sum\nolimits_{l=0}^{\lambda-1}\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\|^2}_{A_2}\\
&\underbrace{\begin{aligned}&-\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}_t^{(i)}),\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\\&-\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\rangle\end{aligned}}_{A_3},
\end{aligned} \tag{31}$$

where the last step is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})$ into $\nabla\tilde{F}_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})$. Below we bound the three terms in (31). For A_1,

$$\begin{aligned}
A_1&=-\tilde{\eta}\sum\nolimits_{i=1}^{I}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}_t^{(i)}),\nabla F(\boldsymbol{w}_t^{(i)})-\nabla F(\boldsymbol{w}_t^{(i)})\\
&+\frac{1}{K\lambda}\sum\nolimits_{k=1}^{K}\sum\nolimits_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\rangle\\
&\overset{(a)}{=}-\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2+\tilde{\eta}\sum\nolimits_{i=1}^{I}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}_t^{(i)}),\\
&\nabla F(\boldsymbol{w}_t^{(i)})-\frac{1}{K\lambda}\sum\nolimits_{k=1}^{K}\sum\nolimits_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}_{k,t-\tau_{k,t}^{(i)},l}^{(i)})\Big\rangle\\
&\overset{(b)}{\leq}-\frac{1}{2}\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2
\end{aligned}$$

$$+\frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}_t^{(i)})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big)\Big\|^2, \tag{32}$$

where (a) is due to $\|\nabla F(\boldsymbol{w}_t)\|^2=\sum_{i=1}^{I}\|\nabla F(\boldsymbol{w}_t^{(i)})\|^2$, (b) follows the triangle inequality. For the last term on the RHS of (32), we have

$$\begin{aligned}
&\frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}_t^{(i)})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big)\Big\|^2\\
&\overset{(a)}{\leq}\frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\nabla F_k(\boldsymbol{w}_t^{(i)})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\|^2\\
&\overset{(b)}{\leq}\tilde{\eta}\sum\nolimits_{i=1}^{I}\frac{1}{K}\sum\nolimits_{k=1}^{K}\mathbb{E}\Big\|\nabla F_k(\boldsymbol{w}_t^{(i)})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})\Big\|^2\\
&+\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\|^2\\
&\overset{(c)}{\leq}\tilde{\eta}\sum\nolimits_{i=1}^{I}\frac{1}{K}\sum\nolimits_{k=1}^{K}L^2\mathbb{E}\Big\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2\\
&+\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}L^2\mathbb{E}\Big\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2,
\end{aligned}\tag{33}$$

where (a) follows Jensen's inequality, (b) comes from adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F_k(\boldsymbol{w}_t^{(i)})$, (c) is due to the L-smooth of loss functions in Assumption 1. Substituting (33) into (32), we have

$$\begin{aligned}
A_1\leq&-\frac{1}{2}\tilde{\eta}\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2+\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\Big\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2\\
&+\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2.
\end{aligned}\tag{34}$$

Now we focus on bounding A_2 as follows:

$$\begin{aligned}
A_2=&\frac{L}{2}\tilde{\eta}^2\sum\nolimits_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum\nolimits_{k=1}^{K}\sum\nolimits_{l=0}^{\lambda-1}\Big(\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\\
&-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})+\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big)\Big\|^2\\
&\overset{(a)}{\leq}\frac{L}{2}\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2\\
&\overset{(b)}{\leq}L\tilde{\eta}^2\sum\nolimits_{i=1}^{I}\mathbb{E}\|\nabla F(\boldsymbol{w}_t^{(i)})\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2\\
&\underbrace{+L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t})\Big)\Big\|^2}_{A_{2,2}},
\end{aligned}\tag{35}$$

where (a) follows the triangle inequality and the bounded noise of SGD in Assumption 3, (b) is derived by adding and subtracting $\nabla F(\boldsymbol{w}_t^{(i)})$ into $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$, then using the triangle inequality. Now we bound the second term on the RHS of (35) as

$$\begin{aligned}
A_{2,2}\overset{(a)}{\leq}&\,2L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\Big(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\\
&-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})\Big)\Big\|^2\\
&+2L\tilde{\eta}^2\sum_{i=1}^{I}\mathbb{E}\Big\|\frac{1}{K}\sum_{k=1}^{K}\Big(\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t})\Big)\Big\|^2\\
&\overset{(b)}{\leq}2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2\\
&+2\tilde{\eta}^2L^3\sum\nolimits_{i=1}^{I}\frac{1}{K}\sum\nolimits_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}-\boldsymbol{w}_t^{(i)}\|^2,
\end{aligned}\tag{36}$$

where (a) is derived by adding and subtracting $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})$, (b) follows Assumption 1. Thus,

$$\begin{aligned}
A_2\leq&\,L\tilde{\eta}^2\sum\nolimits_{i=1}^{I}\mathbb{E}\|\nabla F(\boldsymbol{w}_t^{(i)})\|^2\\
&+2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2\\
&+2\tilde{\eta}^2L^3\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}-\boldsymbol{w}_t^{(i)}\|^2+\frac{L}{2}\tilde{\eta}^2\sigma^2.
\end{aligned}\tag{37}$$

For A_3, we have

$$\begin{aligned}
A_3\overset{(a)}{=}&-\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\langle\nabla F(\boldsymbol{w}_t^{(i)})-\nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}),\\
&\nabla\tilde{F}_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})-\nabla F_k(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l})\Big\rangle\\
&\overset{(b)}{\leq}\frac{1}{2}\tilde{\eta}\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\Big\|\nabla F(\boldsymbol{w}_t^{(i)})-\nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})\Big\|^2+\frac{1}{2}\tilde{\eta}\sigma^2\\
&\overset{(c)}{\leq}\frac{1}{2}\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\Big\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2+\frac{1}{2}\tilde{\eta}\sigma^2,
\end{aligned}\tag{38}$$

where (a) is derived by adding and subtracting $\nabla F(\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}})$ into $\nabla F(\boldsymbol{w}_t^{(i)})$, then using Assumption 3, (b) follow the triangle inequality and the bounded noise of SGD, (c) is due to the L-smooth of loss functions. Substituting (34), (37), (38) into (31), and let $\tilde{\eta}<\frac{1}{2L}$, we have

$$\begin{aligned}
\mathbb{E}\left[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)\right]\leq&(-\frac{1}{2}\tilde{\eta}+L\tilde{\eta}^2)\mathbb{E}\left\|\nabla F(\boldsymbol{w}_t)\right\|^2\\
&+\frac{5}{2}\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\Big\|\boldsymbol{w}_t^{(i)}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2+\frac{3}{4}\tilde{\eta}\sigma^2\\
&+2\tilde{\eta}L^2\sum_{i=1}^{I}\frac{1}{K\lambda}\sum_{k=1}^{K}\sum_{l=0}^{\lambda-1}\mathbb{E}\Big\|\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t},l}-\boldsymbol{w}^{(i)}_{k,t-\tau^{(i)}_{k,t}}\Big\|^2.
\end{aligned}\tag{39}$$

Based on Lemma 1 and Lemma 2, by substituting (10) and (11) into the above inequality and assuming $\tilde{\eta}<\frac{1}{2L}$,

$$\mathbb{E}[F(\boldsymbol{w}_{t+1})-F(\boldsymbol{w}_t)]\leq(-\frac{1}{2}\eta+L\eta^2\lambda)\lambda\left\|\nabla F(\boldsymbol{w}_t)\right\|^2$$

$$+ c_1 + \frac{15\eta^2 L}{4} c_2 \frac{1}{K} \sum\nolimits_{k=1}^{K} \sum\nolimits_{i=1}^{I} (\tau_{k,t}^{(i)})^2, \quad (40)$$

where $c_1 = 4\eta(\lambda-1)IG^2 + \left(4\eta^2 L(\lambda-1)I + \frac{3}{4}\eta\lambda\right)\sigma^2$ and $c_2 = \lambda^2(\sigma^2+G^2) + (\lambda-1)I(\sigma^2+2\lambda G^2)$. According to the evolution of the AoI of local gradients, we have

$$\begin{aligned}(\tau_{k,t}^{(i)})^2 &= (1-\alpha_{k,t}m_{k,t}^{(i)})^2(\tau_{k,t-1}^{(i)}+1)^2 \\ &= (1-\alpha_{k,t}m_{k,t}^{(i)})(\tau_{k,t-1}^{(i)}+1)^2. \end{aligned} \quad (41)$$

Substituting the (41) into (40), the proof is completed.

D. Proof of Corollary 1

By the μ-strongly convex of loss functions, we have $\|\nabla F(\boldsymbol{w}_t)\|^2 \geq 2\mu(F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*))$. Substituting this inequality into (12), then adding and subtracting $F(\boldsymbol{w}^*)$ into (12), we have

$$\begin{aligned}&\mathbb{E}[F(\boldsymbol{w}_{t+1}) - F(\boldsymbol{w}^*)] \\ &\quad \leq (1-\eta\lambda\mu + 2L\eta^2\lambda^2\mu)\mathbb{E}[F(\boldsymbol{w}_t) - F(\boldsymbol{w}^*)] + c_1 \\ &\qquad + \frac{15}{4}\eta^2 L c_2 \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{I}(\tau_{k,t}^{(i)})^2. \end{aligned} \quad (42)$$

By telescoping the above inequality, the proof is completed.

REFERENCES

[1] Z. Chen, W. Yi, L. Sangarapillai, and A. Nallanathan, "Efficient wireless federated learning with adaptive model pruning," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2023.

[2] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 2021.

[3] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33–41, 2022.

[4] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for B5G networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 9–39, 2023.

[5] Z. Chen, W. Yi, Y. Deng, and A. Nallanathan, "Device scheduling for wireless federated learning with latency and representativity," in *Proc.Int. Conf. Electrical, Computer, Commun. and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–6.

[6] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Knowledge-aided federated learning for energy-limited wireless networks," *IEEE Trans. Commun.*, vol. 71, no. 6, pp. 3368–3386, 2023.

[7] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen, "Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking," in *Proc. ACM Conf. Embedded Networked Sensor Systems*, 2021, pp. 42–55.

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

[9] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, 2021.

[10] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, 2021.

[11] Y. Oh, N. Lee, Y.-S. Jeon, and H. V. Poor, "Communication-efficient federated learning via quantized compressed sensing," *IEEE Trans. Wireless Commun.*, vol. 22, no. 2, pp. 1087–1100, 2023.

[12] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, 2020.

[13] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless quantized federated learning: A joint computation and communication design," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2756–2770, 2023.

[14] A. R. Elkordy and A. S. Avestimehr, "Heterosag: Secure aggregation with heterogeneous quantization in federated learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2372–2386, 2022.

[15] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, pp. 1–8, 2022.

[16] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis, and S.-L. Kim, "Mix2fld: Downlink federated learning after uplink federated distillation with two-way mixup," *IEEE Commun. Letters*, vol. 24, no. 10, pp. 2211–2215, 2020.

[17] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu, "Joint model pruning and device selection for communication-efficient federated edge learning," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 231–244, 2022.

[18] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout-a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Commun. Letters*, vol. 11, no. 5, pp. 923–927, 2022.

[19] S. Liu, G. Yu, R. Yin, and J. Yuan, "Adaptive network pruning for wireless federated learning," *IEEE Wireless Commun. Letters*, vol. 10, no. 7, pp. 1572–1576, 2021.

[20] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[21] E. Diao, J. Ding, and V. Tarokh, "Heterofl: Computation and communication efficient federated learning for heterogeneous clients," *arXiv preprint arXiv:2010.01264*, 2020.

[22] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Trans. Neural Networks and Learning Systems*, pp. 1–13, 2022.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[24] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2017.

[25] F. Ilhan, G. Su, and L. Liu, "Scalefl: Resource-adaptive federated learning with heterogeneous clients," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 24 532–24 541.

[26] P. Li, G. Cheng, X. Huang, J. Kang, R. Yu, Y. Wu, and M. Pan, "Anycostfl: Efficient on-demand federated learning over heterogeneous edge devices," *arXiv preprint arXiv:2301.03062*, 2023.

[27] S. Lin, R. Ji, Y. Li, C. Deng, and X. Li, "Toward compact convnets via structure-sparsity regularized filter pruning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 574–588, 2020.

[28] Y. Mei, P. Guo, M. Zhou, and V. Patel, "Resource-adaptive federated learning with all-in-one neural composition," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 4270–4284.

[29] Z. Chen, W. Yi, and A. Nallanathan, "Exploring representativity in device scheduling for wireless federated learning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2023.

[30] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *Proc. Int. Conf. Learning Representations (ICLR)*, April, 2017.

[31] Z. Chen, W. Yi, A. Nallanathan, and G. Y. Li, "Efficient wireless federated learning with partial model aggregation," *arXiv preprint arXiv:2204.09746*, 2022.

[32] A. Schrijver *et al.*, *Combinatorial optimization: polyhedra and efficiency*. Springer, 2003, vol. 24.

[33] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," *J. ACM*, vol. 68, no. 1, jan 2021.

[34] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, 2022.

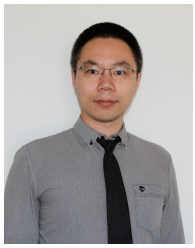
[35] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2015.

[36] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless iot networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3394–3409, 2021.

[37] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

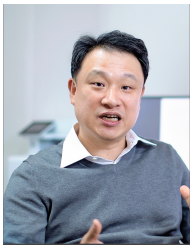
Zhixiong Chen received the B.S. and M.S. degrees from Chongqing University, Chongqing, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering and Computer Science, Queen Mary University of London.

His research interests include machine learning in wireless networks, reinforcement learning, and wireless federated learning.



Wenqiang Yi (S'17-M'20) received his Ph.D. degree in electrical engineering from Queen Mary University of London, U.K., in 2020. He is currently an Assistant Professor in the School of Computer Science and Electronic Engineering, University of Essex, since 2023. From 2020 to 2023, he was a Post-Doctoral Researcher with Communication Systems Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London.

His research interests include AI in wireless communications, RF sensing, and stochastic geometry. He serves as an Associate Editor for IEEE OJ-COMS, in the area of Big Data and Machine Learning for Communications. He received the Exemplary Reviewer of the IEEE COMMUNICATION LETTERS and the IEEE TRANSACTIONS ON COMMUNICATIONS in 2019 and 2020. He served as the symposium chair on reconfigurable intelligent surfaces at IEEE ICCT. He has served as a TPC Member for many IEEE conferences, e.g., GLOBECOM and ICC. He also served as the Secretary of the Special Interest Group on Next Generation Multiple Access (NGMA) by the SPCC Technical Committee and the Emerging Technologies Initiatives on NGMA by the Emerging Technologies Committee till 2022.



Hyundong Shin (Fellow, IEEE) received the B.S. degree in Electronics Engineering from Kyung Hee University (KHU), Yongin-si, Korea, in 1999, and the M.S. and Ph.D. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 2001 and 2004, respectively. During his post-doctoral research at the Massachusetts Institute of Technology (MIT) from 2004 to 2006, he was with the Laboratory for Information Decision Systems (LIDS). In 2006, he joined the KHU, where he is currently a Professor in the Department of Electronic Engineering. His research interests include quantum information engineering, wireless communication, and machine intelligence. Dr. Shin received the IEEE Communications Society's Guglielmo Marconi Prize Paper Award and William R. Bennett Prize Paper Award. He served as the Publicity Co-Chair for the IEEE PIMRC and the Technical Program Co-Chair for the IEEE WCNC and the IEEE GLOBECOM. He was an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS.

Arumugam Nallanathan (S'97-M'00-SM'05-F'17) is Professor of Wireless Communications and Head of the Communication Systems Research (CSR) group in the School of Electronic Engineering and Computer Science at Queen Mary University of London since September 2017. He was with the Department of Informatics at Kings College London from December 2007 to August 2017, where he was Professor of Wireless Communications from April 2013 to August 2017 and a Visiting Professor from September 2017 to August 2020. He was an Assistant Professor in the Department of Electrical and Computer Engineering, National University of Singapore from August 2000 to December 2007. His research interests include Artificial Intelligence for Wireless Systems, Beyond 5G Wireless Networks and Internet of Things (IoT). He published nearly 700 technical papers in scientific journals and international conferences. He is a co-recipient of the Best Paper Awards presented at the IEEE International Conference on Communications 2016 (ICC'2016), IEEE Global Communications Conference 2017 (GLOBECOM'2017) and IEEE Vehicular Technology Conference 2018 (VTC'2018). He is also a co-recipient of IEEE Communications Society Leonard G. Abraham Prize in 2022. He is an IEEE Distinguished Lecturer. He has been selected as a Web of Science Highly Cited Researcher in 2016, 2022 and 2023.

He is a Senior Editor for IEEE Wireless Communications Letters. He was an Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Communications, IEEE Transactions on Vehicular Technology and IEEE Signal Processing Letters. He served as the Chair for the Signal Processing and Communication Electronics Technical Committee of IEEE Communications Society and Technical Program Chair and member of Technical Program Committees in numerous IEEE conferences. He received the IEEE Communications Society SPCE outstanding service award 2012 and IEEE Communications Society RCC outstanding service award 2014.