1

Knowledge-aided Federated Learning for Energy-limited Wireless Networks

Zhixiong Chen, *Student Member, IEEE*, Wenqiang Yi, *Member, IEEE*,

Yuanwei Liu, *Senior Member, IEEE*, and Arumugam Nallanathan, *Fellow, IEEE*

Abstract

The conventional model aggregation-based federated learning (FL) approach requires all local models to have the same architecture, which fails to support practical scenarios with heterogeneous local models. Moreover, the frequent model exchange is costly for resource-limited wireless networks since modern deep neural networks usually have over a million parameters. To tackle these challenges, we first propose a novel knowledge-aided FL (KFL) framework, which aggregates light high-level data features, namely knowledge, in the per-round learning process. This framework allows devices to design their machine-learning models independently and reduces the communication overhead in the training process. We then theoretically analyze the convergence bound of the proposed framework under a non-convex loss function setting, revealing that scheduling more data volume in each round helps to improve the learning performance. In addition, large data volume should be scheduled in early rounds if the total scheduled data volume during the entire learning course is fixed. Inspired by this, we define a new objective function, i.e., the weighted scheduled data sample volume, to transform the inexplicit global loss minimization problem into a tractable one for device scheduling, bandwidth allocation, and power control. To deal with unknown time-varying wireless channels, we transform the considered problem into a deterministic problem for each round with the assistance of the Lyapunov optimization framework. Then, we derive the optimal bandwidth allocation and power control solution by convex optimization techniques. We also develop an efficient online device scheduling algorithm to achieve an energy-learning trade-off in the learning process. Experimental results on two typical datasets (i.e., MNIST and CIFAR-10) under highly heterogeneous local data distributions show that the proposed KFL is capable of reducing over 99% communication overhead while achieving better learning performance than the conventional model aggregation-based algorithms. In addition, the proposed device scheduling algorithm converges faster than the benchmark scheduling schemes.

Index Terms

Device scheduling, Lyapunov optimization, personalized federated Learning, resource allocation

Zhixiong Chen, Wenqiang Yi, Yuanwei Liu, and Arumugam Nallanathan are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. (emails: {zhixiong.chen, w.yi, yuanwei.liu, a.nallanathan}@qmul.ac.uk).

2

I. INTRODUCTION

The increasing demands for intelligent services, such as augmented reality/virtual reality (AR/VR) and Internet-of-Things (IoT) applications, motivate the integration of machine learning in future wireless networks [2]. Federated learning (FL) is one of the most promising distributed learning frameworks to reduce the communication traffic load of intelligent services, which enables devices to collaboratively train machine learning models by periodically exchanging model parameters between devices and the parameter server instead of raw user data [3]. However, the model aggregation nature of conventional FL confronts the following two limitations for its implementation in wireless networks: 1) *High Communication Overhead:* The uploading of model/gradient parameters is costly for devices since modern deep neural network (NN) architectures usually possess massive parameters. For instance, the widely used MobileNet [4], a convolutional NN (CNN) for on-device image processing, has 6.9 million parameters, corresponding to 27.6 MB. Training such a model requires devices to upload 27.6 MB of data per round. Considering hundreds of rounds and multiple devices, the communication overhead is heavy for wireless networks with limited spectrum and energy resources. 2) *Heterogeneous Local Models:* In practical wireless networks, devices are usually equipped with different NNs in terms of architectures and model sizes due to their heterogeneous computing capabilities and storage resources [5]. In this case, the traditional model aggregation-based FL approaches fail to coordinate devices to perform the learning process. To break these two limitations, state-of-the-art studies focus on the designs of communication-efficient FL and heterogeneous FL.

A. Related Works

To enable communication-efficient FL in resource-limited wireless networks, existing works mainly focused on device scheduling [6]–[9], model quantization [10]–[13], and model pruning [14]–[16]. Device scheduling methods select a small subset of devices to participate in the per-round training process, thus reducing the communication burden and mitigating the straggler effect when devices have random or heterogeneous computing speed. The device selection and bandwidth allocation in [6], [7] guaranteed long-term learning performance in bandwidth-limited wireless networks. The probabilistic scheduling policy for FL proposed in [8] effectively minimized the model uploading latency and improved convergence speed. The co-design of learning and device selection in [9] reduced convergence time in resource-constrained wireless networks. Although these device scheduling approaches efficiently alleviate communication burden, trans-

mitting the entire model is arduous for devices with weak channels and limited energy. To tackle this issue, the model quantization compresses devices' model updates before transmitting to the parameter server, thus reducing the transmitted data volume and communication overhead for devices [10], [11]. Specifically, the model quantization approach in [12] enabled edge devices to adjust their quantization proportional according to their communication resources for balancing training accuracy and communication overhead. The heterogeneous quantization method in [13] allocated different aggregation weights to clients for efficiently improving convergence speed. While the model quantization is demonstrably effective, it introduces additional noise during training, which ultimately degrading the trained model's performance. The model pruning is able to simultaneously reduce communication and computation costs by removing less important weights from the original model. The joint design of the pruning ratio and wireless resource allocation in [14] significantly improved the convergence rate of FL. In [15], the NN pruning was integrated into FL to improve learning speed and guarantee training latency. A random model pruning approach was adopted in [16] to generate several subnets from the global model to adapt the channel condition of different devices. It reduced both communication overhead and computation loads. The above three approaches reduce communication overhead while degrading the final model's accuracy. Besides these approaches, our previous work [17] enabled devices to train the feature extractor part of NNs collaboratively, while the predictor part for devices is localized for personalization. It reduced communication overhead and improved learning performance in heterogeneous data distribution scenarios. However, these approaches still require heavy parameter transmission in the learning process.

To allow devices equipped with heterogeneous models in FL, knowledge distillation (KD)-based FL approaches were developed and attracted much attention. In practical wireless networks, devices usually possess different computation capabilities and communication resources. Thus, requiring all the local models to be of the same architecture in many application scenarios may be ineffective. KD is a teacher-student paradigm which transfers the knowledge distilled from the teacher model to the student model [18]. Integrating KD into FL allows devices to independently design their models according to channel conditions and computation capabilities. Specifically, the federated KD approach in [19] effectively enabled federated training between heterogeneous models by aggregating local models' logits on a public dataset. In [20], an auxiliary distillation dataset generated by mixing local training data was adopted to empower the FL process, effectively reducing convergence time. In [21], a lightweight generator was deployed

at the server to ensemble user information and broadcast to devices to regulate their local training process. By deploying an unlabelled dataset on both the server and devices, a global model was trained using the averaged outputs of local models on this dataset as the supervision label [22], [23]. The adaptive mutual KD and dynamic gradient compression approach in [24] significantly reduced communication costs and achieved competitive results with centralized model learning. The federated distillation method [25] regularized local models to mitigate overfitting during training by treating the global model as the teacher and the local models as the students. Besides enabling devices to design their machine learning models independently, the KD-based FL substantially reduces the transmitted data volume in the wireless channels because output logits are required to upload in the learning process instead of heavy model/gradient parameters. However, these KD-based FL approaches require an extra public dataset to align the student and teacher models' outputs, increasing the computation costs. Moreover, their performance may significantly degrade with the increase in the distribution divergence between the public and on-device datasets that are usually non-independent and identically distributed (non-IID).

B. Motivations and Contributions

Although the communication-efficient FL in [6]–[16] can reduce communication overhead, they degrade the final model accuracy and require heavy parameter transmission in the learning process. In addition, the KD-based FL in [19]–[25] allowed devices to ensemble heterogeneous local models. However, they rely on the public dataset, which may not be practical for many scenarios. To break these limitations, this work aims to enable collaborative training for devices equipped with heterogeneous models in a communication-efficient way, avoiding the reliance on heavy model transmission and extra public datasets. Inspired by the human experience in discriminating between different objects and its successful application in clustering analysis, the same class of objects or data usually have similar high-level features, while different objects have distinct features [26]. In addition, a general insight for modern deep learning models is that the lower layers (close to the input) are primarily responsible for feature extraction, while the upper layers (proximate to the output) focus on complex pattern recognition [27]. We aim to enable devices for collaborative training by aggregating their output of lower layers of the NNs, namely knowledge, in the per-round training process. This design effectively reduces the communication overhead since the dimensions of the knowledge are usually much smaller than that of the model. The main contributions of this paper are summarized as follows:

- We propose a novel KFL framework in which devices collaboratively train models by uploading their knowledge of different data classes to the edge server for aggregation. This design reduces the transmitted data volume in the wireless channels, allowing devices to design their machine-learning models independently according to their computation capabilities and communication conditions.

- We theoretically analyze the convergence bound of the proposed KFL framework under the general non-convex loss function setting, which indicates that scheduling more data samples in each round is able to improve the learning performance. In addition, when the total number of scheduled data volume during the entire learning course is fixed, more data volume should be scheduled in the early rounds. Following the experimental investigation of temporal scheduling policies in [6], this work further theoretically analyzes how the temporal device scheduling patterns affect the final learning performance through the convergence analysis.

- We formulate a long-term device scheduling, bandwidth allocation, and power control problem under limited devices' energy budgets with the aid of the convergence bound. To deal with unpredicted time-varying wireless channels and enable online device scheduling, we first transform the original problem into a deterministic problem in each round with the assistance of the Lyapunov optimization framework. Then, we derive the optimal bandwidth allocation and power control through convex optimization techniques. Finally, we develop an efficient polynomial-time algorithm to solve the device scheduling policy with $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off guarantee, where V is an algorithm-specific parameter.

- We experimentally verify the correctness of our theoretical results, i.e., more data samples should be scheduled in the early rounds when the total scheduled data volume in the entire learning course are fixed. Compared with benchmark FL algorithms, the proposed KFL framework saves 99% communication overhead and boosts 2.1% and 6.65% accuracy on MNIST and CIFAR-10 datasets, respectively. In addition, The proposed online device scheduling algorithm achieves a faster convergence speed than benchmark scheduling approaches.

C. Organization and Notations

The rest of this paper is organized as follows: In Section II, we introduce the proposed KFL system and learning cost, then formulate the global loss minimization problem. The convergence
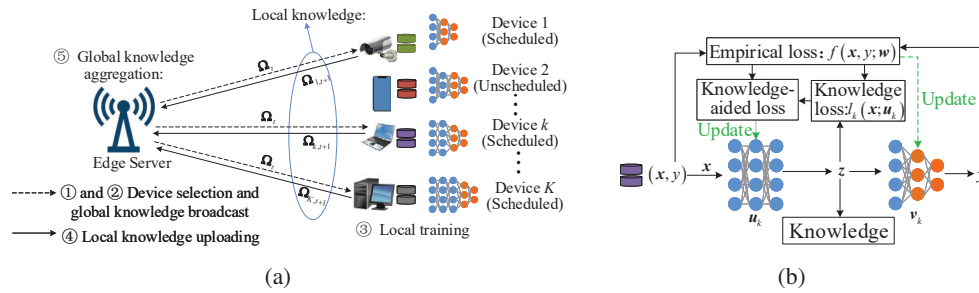
Fig. 1. The illustrated KFL over wireless networks: (a) Federated learning with knowledge aggregation mechanism, where devices have different local models; and (b) Local training process with the proposed knowledge-aided loss.

analysis and problem transformation are illustrated in Section III. The joint device scheduling, bandwidth allocation, and power control algorithm are developed in IV. Section V verifies the effectiveness of the proposed scheme by simulation. The conclusion is drawn in Section VI. For convenience, we use "$\triangleq$" to denote "is defined to be equal to", $|\cdot|$ denote the size operation of a set, $\nabla(\cdot)$ denote gradient operator, $\langle\cdot,\cdot\rangle$ denote inner product operator, and "$\|\cdot\|$" denote the ℓ_2 norm throughout this paper. The main notations used in this paper are summarized in Table I.

TABLE I
NOTATION SUMMARY

Notation	Definition	Notation	Definition
$\mathcal{K}$; K;	Set of devices; size of $\mathcal{K}$	$\mathcal{C}$; C;	Set of classes; size of $\mathcal{C}$
$\mathcal{D}_k$; D_k	Local dataset of device k; size of $\mathcal{D}_k$	$\mathcal{D}$; D	Overall dataset in the system; size of $\mathcal{D}$
$\mathcal{D}_{k,c}$; $D_{k,c}$	Local dataset of c class; size of $\mathcal{D}_{k,c}$	$\boldsymbol{w}_k$; $\boldsymbol{u}_k$; $\boldsymbol{v}_k$; $\boldsymbol{W}$	Local model; local feature extractor; local predictor of device k; all local models
$F_k(\boldsymbol{u}_k,\boldsymbol{v}_k)$; $F(\boldsymbol{W})$	Local empirical loss function of device k; global empirical loss function	η_u; η_v	Learning rate for feature extractor and predictor
$L_k(\boldsymbol{u}_k)$	Local knowledge loss function	λ	Knowledge loss weight
$\boldsymbol{\Omega}_{k,c}$; $\boldsymbol{\Omega}_k$	Device k's knowledge about class c; device k's knowledge for all classes	$\boldsymbol{\Omega}_c$; $\boldsymbol{\Omega}$	Global knowledge about class c; global knowledge about all classes
$\boldsymbol{S}_t$; τ	Scheduling policy in round t, i.e., the set of scheduled devices; local iteration number	f_k; C_k	CPU frequency of device k; Computation workload of one data sample at device k
$p_{k,t}$; $p_{k,\max}$	Transmit power of device k in round t; maximum transmit power of device k	C_k; Q	Computation workload of one data sample at device k; Data size of local knowledge
B; $\boldsymbol{\theta}_t$	Wireless bandwidth; the proportion of B allocated to devices in round t	E_k; $\mathcal{T}_{\max}$	Total energy budget of device k; Maximum completion time for each round

II. SYSTEM MODEL AND LEARNING MECHANISM

In the considered KFL system, as shown in Fig. 1, an edge server coordinates K different devices to train machine learning models for classification or recognition tasks. Unlike the conventional FL that requires all devices' models to be of the same architecture, the KFL in this work allows devices to be equipped with heterogeneous models. The devices are indexed by $\mathcal{K} = \{1, 2, \cdots, K\}$. For the dataset at devices, the number of data classes in the classification or recognition task is C, indexed by $\mathcal{C} = \{1, 2, \cdots, C\}$. Each device k ($k \in \mathcal{K}$) has a local dataset $\mathcal{D}_k$ with $D_k = |\mathcal{D}_k|$ data samples, in which the data samples belong to c-th class ($c \in \mathcal{C}$) is denoted as $\mathcal{D}_{k,c}$ with $D_{k,c} = |\mathcal{D}_{k,c}|$ data samples. Thus, $\mathcal{D}_k = \cup\{\mathcal{D}_{k,c}\}_{c=1}^{C}$. Without loss of generality, we assume there is no overlapping between datasets from different devices, i.e., $\mathcal{D}_k \cap \mathcal{D}_h = \emptyset$, $\forall k, h \in \mathcal{K}$. Thus, the entire dataset, $\mathcal{D} = \cup\{\mathcal{D}_k\}_{k=1}^{K}$, is with total number

of samples $D = \sum_{k=1}^{K} D_k$. For ease of presentation, we use $\mathcal{D}_c$ to represent all data samples belonging to class c in $\mathcal{D}$. That is, $\mathcal{D}_c = \cup \{\mathcal{D}_{k,c}\}_{k=1}^{K}$ with $D_c = \sum_{k=1}^{K} D_{k,c}$ data samples.

A. Knowledge-aided Loss Function for Local Training

Let $\zeta = (\boldsymbol{x}, y)$ denote a data sample in $\mathcal{D}$, where $\boldsymbol{x} \in \mathbb{R}^d$ is the d-dimensional input feature vector, $y \in \mathbb{R}$ is the corresponding ground-truth label. Let $\boldsymbol{z} \in \mathbb{R}^p$ be the latent feature vector. As shown in Fig. 1(b), the machine learning model parameterized by $\boldsymbol{w} = [\boldsymbol{u}, \boldsymbol{v}]$ consists of two components: a feature extractor $h : \boldsymbol{x} \to \boldsymbol{z}$ parameterized by $\boldsymbol{u}$, and a label predictor $g : \boldsymbol{z} \to \hat{y}$ parameterized by $\boldsymbol{v}$. Before discussing the knowledge-aided loss function, we introduce two fundamental loss functions, i.e., empirical loss and knowledge loss. The empirical loss supervises the local models' training to minimize the prediction error, while the knowledge loss achieves knowledge sharing among devices.

1) **Empirical loss function for local model update**: Let $f(\boldsymbol{x}, y; \boldsymbol{w})$ denote the sample-wise empirical loss function, which quantifies the error between the ground-truth label, y, and the predicted output, $\hat{y}$, based on model $\boldsymbol{w}$. Thus, the local empirical loss function at device k, which measures the model error on its local dataset $\mathcal{D}_k$, is defined as

$$F_k(\boldsymbol{w}_k) = F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \triangleq \frac{1}{D_k} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_k} f(\boldsymbol{x}, y; \boldsymbol{w}_k), \tag{1}$$

where $\boldsymbol{w}_k$ denotes the machine learning model of device k; $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ correspond to its feature extractor and label predictor parts, respectively. For ease of presentation, we use $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K)$ to denote all the devices' models throughout this paper. The global loss function associated with all distributed local datasets is given by

$$F(\boldsymbol{W}) = F(\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K) \triangleq \frac{1}{D} \sum_{k=1}^{K} D_k F_k(\boldsymbol{w}_k). \tag{2}$$

The federated learning process is done by solving the following problem:

$$\min_{\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_K)} F(\boldsymbol{W}). \tag{3}$$

To preserve the data privacy of devices, the devices collaboratively learn $\boldsymbol{W}$ without transmitting the raw training data. Note that the conventional FL algorithms, e.g., FedAvg [28], aim to find an optimal shared global model $\boldsymbol{w}^* = \boldsymbol{w}_1^* = \cdots = \boldsymbol{w}_K^*$ to minimize the global loss $F(\boldsymbol{W})$. However, this work aims to develop a personalized FL algorithm which trains personalized models for each device to solve the problem (3), where different local models are used to fit user-specific data and capture the common knowledge distilled from data of other devices.

2) **Knowledge loss function for local feature extractor update**: When devices are equipped with heterogeneous models, the conventional FL algorithms fail to coordinate devices to train

models collaboratively. To tackle this issue, we introduce the knowledge loss function to regularize devices' feature extractors in the training process, achieving knowledge sharing between devices. It is worth mentioning that the knowledge of different devices and classes has the same dimensionality that equals the dimension of feature extractors' output, i.e., p. Let $\boldsymbol{\Omega}_{k,c}$ denote device k's knowledge about data class c, which is defined as the average output of its feature extractor based on the data samples in $\mathcal{D}_{k,c}$, that is

$$\boldsymbol{\Omega}_{k,c} = \frac{1}{D_{k,c}} \sum\nolimits_{(\boldsymbol{x},y) \in \mathcal{D}_{k,c}} h_k(\boldsymbol{x}; \boldsymbol{u}_{k,t}), \tag{4}$$

where $h_k(\cdot)$ denote the feature extractor of device k. Let $\boldsymbol{\Omega}_c$ denote the global knowledge about class c that aggregates all devices' knowledge of class c, i.e.,

$$\boldsymbol{\Omega}_c = \frac{1}{D_c} \sum\nolimits_{k=1}^{K} D_{k,c} \boldsymbol{\Omega}_{k,c}. \tag{5}$$

We use $\boldsymbol{\Omega} = (\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \cdots, \boldsymbol{\Omega}_C)$ to denote the aggregated global knowledge. For each data sample $(\boldsymbol{x}, y) \in \mathcal{D}_{k,c}$ ($\forall k \in \mathcal{K}, c \in \mathcal{C}$), we define the knowledge loss of device k's feature extractor as $l_k(\boldsymbol{x}; \boldsymbol{u}_k) = \frac{1}{2}\|h_k(\boldsymbol{x}; \boldsymbol{u}_k) - \boldsymbol{\Omega}_c\|^2$, which quantifies the difference between the extracted feature of device k on data sample $(\boldsymbol{x}, y)$ and the global feature of class c. Thus, the knowledge loss of device k is

$$L_k(\boldsymbol{u}_k) = \frac{1}{D_k} \sum\nolimits_{c=1}^{C} \sum\nolimits_{(\boldsymbol{x},y) \in \mathcal{D}_{k,c}} \frac{1}{2}\|h_k(\boldsymbol{x}; \boldsymbol{u}_{k,t}) - \boldsymbol{\Omega}_c\|^2, \tag{6}$$

which measures the difference between local knowledge and global knowledge. According to (6), devices only learn the knowledge of their local data types instead of all the data types. However, it fits devices' local models to their specific data and improves the learning performance on heterogeneous local data scenarios. In addition, devices can use global knowledge to regularize the local training process when new data classes are generated and rapidly adapt their local models to these new class data.

In this work, we define a **knowledge-aided loss function** based on the empirical and knowledge loss functions, i.e., $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) + \lambda L_k(\boldsymbol{u}_k)$, to guide the feature extractor training for device k ($\forall k \in \mathcal{K}$), where λ is a hyperparameter to balance the empirical loss and knowledge loss for device k. For the label predictor, we still use the conventional empirical loss function.

B. Knowledge-aided Federated Learning Mechanism

The conventional FL approaches rely on aggregating devices' model/gradient parameters in each round, which induces remarkable communication overhead for wireless networks and requires all the local models to be of the same architecture. To tackle these issues, we propose a novel KFL algorithm to enable collaborative training between heterogeneous local models.

Specifically, devices upload their lightweight *knowledge* to the server for aggregation in the per-round training process instead of the heavy model/gradient parameters. The learning process repeats the following steps until the devices' models converge, as shown in Fig. 1(a).

1) **Device selection**: The edge server selects a subset of devices from $\mathcal{K}$ to participate in the training process in the current round. Let $\alpha_{k,t} \in \{0, 1\}$ denote the scheduling indicator of device k in round t, where $\alpha_{k,t} = 1$ indicates that device k is scheduled in round t, $\alpha_{k,t} = 0$ otherwise. Thus, the scheduled device set in round t is $\boldsymbol{S}_t = \{k : \alpha_{k,t} = 1, \forall k \in \mathcal{K}\}$.

2) **Knowledge broadcast**: In each round t, the edge server broadcasts the latest global knowledge, i.e., $\boldsymbol{\Omega}_t = (\boldsymbol{\Omega}_{1,t}, \boldsymbol{\Omega}_{2,t}, \cdots, \boldsymbol{\Omega}_{C,t})$, to all scheduled devices to regularize their local training process, where $\boldsymbol{\Omega}_{c,t}$ is the c-th class knowledge in round t that is computed in (5).

3) **Local training**: All scheduled devices update their local models after receiving the global knowledge, $\boldsymbol{\Omega}_t$, by performing τ steps gradient descent on its local dataset, as shown in Fig. 1(b). For device k, its local feature extractor in t-th round is updated as

$$\boldsymbol{u}_{k,t,l+1} = \boldsymbol{u}_{k,t,l} - \eta_u \Big(\nabla_u F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l}) \Big), \forall l \in \{0, 1, \cdots, \tau - 1\}, \quad (7)$$

and its predictor is updated by

$$\boldsymbol{v}_{k,t,l+1} = \boldsymbol{v}_{k,t,l} - \eta_v \nabla_v F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}), \forall l \in \{0, 1, \cdots, \tau - 1\}, \quad (8)$$

where η_u and η_v are the learning rate of feature extractor and predictor, respectively, λ is a hyperparameter to balance the empirical loss and knowledge loss for devices k.

4) **Knowledge computing**: After finishing the local iterations, all scheduled devices compute their knowledge for each class c ($c \in \mathcal{C}$) as $\boldsymbol{\Omega}_{k,c,t+1} = \frac{1}{D_{k,c}} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t+1}; \boldsymbol{x})$. The knowledge of device k for all classes is denoted by $\boldsymbol{\Omega}_{k,t+1} = (\boldsymbol{\Omega}_{k,1,t+1}, \boldsymbol{\Omega}_{k,2,t+1}, \cdots, \boldsymbol{\Omega}_{k,C,t+1})$.

5) **Knowledge aggregation**: After finishing the local knowledge computing, all scheduled devices upload their knowledge to the edge server through wireless channels for aggregation. Specifically, the edge server computes the global shared knowledge of c-th class as

$$\boldsymbol{\Omega}_{c,t+1} = \frac{\sum_{k \in \boldsymbol{S}_t} D_{k,c} \boldsymbol{\Omega}_{k,c,t+1}}{\sum_{k \in \boldsymbol{S}_t} D_{k,c}}. \quad (9)$$

The aggregated global knowledge in round $(t+1)$ is $\boldsymbol{\Omega}_{t+1} = (\boldsymbol{\Omega}_{1,t+1}, \boldsymbol{\Omega}_{2,t+1}, \cdots, \boldsymbol{\Omega}_{C,t+1})$.

To better illustrate the proposed KFL, we summarize the detailed steps of its training process in Algorithm 1. It is worth mentioning that the proposed KFL requires devices to upload the knowledge to the edge server for aggregation instead of the entire local models. Devices' knowledge is generated by averaging the output of their local feature extractor on the data samples from the same class, and the process is irreversible [29]. Thus, KFL is more beneficial for privacy preservation than the model aggregation-based FL algorithms exchanging local models

between devices and the edge server. The reason is that the local models are updated according to the devices' private data, whose pattern is encoded into the model parameters. Therefore, if a corresponding decoder could be constructed, the private data or statistics would be recovered inversely [30].

Algorithm 1 Knowledge-aided Federated Learning Algorithm

1: **Initialization:** $t = 0$, training round T, and each device initials its local model $\boldsymbol{w}_{k,t}$;
2: **Server side:**
3: **for** $t = 0, 1, \cdots, T - 1$ **do**
4:      Select a subset of devices ($\boldsymbol{S}_t$) and broadcasts the latest global knowledge, i.e., $\boldsymbol{\Omega}_t$, to them.
5:      **if** Receive the knowledge from the selected devices **then**
6:          Aggregate the global knowledge according to (9).
7:      **end if**
8: **end for**
9: **Device side:**
10: **if** Device k is scheduled **then**
11:      Receive the global knowledge, $\boldsymbol{\Omega}_t$, from the edge server;
12:      **for** $l = 0, 1, \cdots, \tau - 1$ **do**
13:          Update the local feature extractor, $\boldsymbol{u}_{k,t,l+1}$, based on (7);
14:          Update the local predictor, $\boldsymbol{v}_{k,t,l+1}$, based on (8);
15:      **end for**
16:      Compute their knowledge for each class c ($c \in \mathcal{C}$) as $\boldsymbol{\Omega}_{k,c,t+1} = \frac{1}{D_{k,c}} \sum_{(\boldsymbol{x},y) \in \mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t+1}; \boldsymbol{x})$.
17:      Upload the local knowledge $\boldsymbol{\Omega}_{k,t+1} = (\boldsymbol{\Omega}_{k,1,t+1}, \boldsymbol{\Omega}_{k,2,t+1}, \cdots, \boldsymbol{\Omega}_{k,C,t+1})$ to the edge server.
18: **end if**

C. Knowledge-aided Federated Learning Cost Model

In the following, we characterize the learning cost model in each KFL round, including computation cost and communication cost.

1) **Computation Cost**: We consider the central processing unit (CPU) adopted to perform training on each device. Denote the CPU clock frequency of device k by f_k (cycles per second). The number of float-point operations (FLOPs) per cycle is represented by n_k. Let C_k denote the required number of FLOPs to process one data sample at device k. Consequently, the local training latency of device k is given by

$$\mathcal{T}_k^{\mathrm{L}} = \tau D_k C_k / (f_k n_k). \tag{10}$$

The corresponding energy consumption of device k is

$$E_k^{\mathrm{L}} = \kappa \tau D_k C_k f_k^2 / n_k, \tag{11}$$

where κ is the power coefficient, depending on the chip architecture.

2) **Communication Cost**: We consider that the frequency division multiple access is employed for devices to upload their knowledge. The total available wireless bandwidth is BHz. Let $p_{k,t}$ denote the transmit power of device k, its maximum value is $p_{k,\mathrm{max}}$. The channel gain between device k and the edge server is represented by $h_{k,t}$, which considers the path loss and Rayleigh fading. In addition, the channel remains unchangeable within one round but varies independently over rounds. Let $\theta_{k,t} \in [0, 1]$ denote the proportion of the overall bandwidth allocated to device k in round t, and $\boldsymbol{\theta}_t = (\theta_{1,t}, \theta_{2,t}, \cdots, \theta_{K,t})$. The uplink rate of device k can be described as

$r_{k,t} = \theta_{k,t} B \log_2(1 + \frac{p_{k,t}h_{k,t}}{\theta_{k,t}BN_0})$, where N_0 is the power density of noise. Note that the proposed KFL requires that the knowledge of different devices and classes has the same dimensionality. Thus, the number of parameters in the knowledge of different devices is the same, denoted as Q. Each parameter is quantized by q bits. Thus, the local knowledge uploading latency of device k is

$$\mathcal{T}_{k,t}^{U} = \frac{Qq}{r_{k,t}} = \frac{Qq}{\theta_{k,t} B \log_2\left(1 + \frac{p_{k,t}h_{k,t}}{\theta_{k,t}BN_0}\right)}. \tag{12}$$

The corresponding energy consumption is

$$E_{k,t}^{U} = p_{k,t}\mathcal{T}_{k,t}^{U} = \frac{\theta_{k,t} B \mathcal{T}_{k,t}^{U} N_0}{h_{k,t}}\left(2^{\frac{Qq}{\theta_{k,t}B\mathcal{T}_{k,t}^{U}}} - 1\right). \tag{13}$$

According to above modes, the energy consumption of device k in round t is $E_{k,t} = E_{k,t}^{L} + E_{k,t}^{U}$. Note that we ignore the global knowledge broadcasting and aggregation latency in the above discussion because the broadcasting process occupies the entire bandwidth. The edge server has large transmit power, so the broadcasting latency is negligible. Moreover, the edge server is usually computationally powerful, and the global knowledge aggregation latency can be ignored compared to the above computation and communication latencies.

D. Problem Formulation

In this work, we aim to improve the learning performance by minimizing the global loss after T rounds, i.e., $F(\boldsymbol{W}_T)$, under the energy budget constraint of devices, where $\boldsymbol{W}_T$ denote the local models in T-th round. Towards this end, we jointly optimize the device scheduling, bandwidth allocation, and power control policies. The optimization problem is given by

$$\mathcal{P}: \min_{\{\boldsymbol{S}_t,\boldsymbol{\theta}_t,\boldsymbol{p}_t\}_{t=0}^{T-1}} F(\boldsymbol{W}_T) \tag{14}$$

$$\text{s. t.} \quad \sum_{t=0}^{T-1} E_{k,t} \leq E_k, \forall k \in \mathcal{K}, \tag{14a}$$

$$\mathcal{T}_{k,t}^{L} + \mathcal{T}_{k,t}^{U} \leq \mathcal{T}_{\max}, \forall k \in \mathcal{K}, \forall t, \tag{14b}$$

$$\sum_{k=1}^{K} \theta_{k,t} \leq 1, \forall t, \tag{14c}$$

$$0 \leq \theta_{k,t} \leq 1, \forall k \in \mathcal{K}, \forall t, \tag{14d}$$

$$\alpha_{k,t} \in \{0, 1\}, \forall k \in \mathcal{K}, \forall t, \tag{14e}$$

$$0 \leq p_k \leq p_{k,\max}, \forall k \in \mathcal{K}. \tag{14f}$$

In problem $\mathcal{P}$, (14a) imposes restrictions on the energy consumption of each device k cannot exceed its budget E_k. (14b) stipulates that the completion time of each round cannot exceed its maximum allowable delay. (14c) indicates that the wireless bandwidth allocated to all devices cannot exceed the total available bandwidth resource. (14d) restricts the wireless bandwidth resource allocated to each device. (14e) indicates which devices are scheduled in each round.

Solving problem $\mathcal{P}$ requires the explicit form about how device scheduling policy affects the final global loss function. Since it is almost impossible to find an exact analytical expression of $F(\boldsymbol{W}_T)$ with respect to $\boldsymbol{S}_t$ ($t \in \{0, 1, \cdots, T-1\}$), we turn to find an upper bound of $F(\boldsymbol{W}_T)$ and minimize it for the global loss minimization in Section III-A. Moreover, the optimal solution to problem $\mathcal{P}$ requires the system state information of all rounds at the beginning of training. However, such information is unavailable in the practical systems due to the unpredictable time-varying channel condition. To enable online device scheduling, the device scheduling decision should be made at the beginning of each round with only the current state. To this end, we transform the long-term decision problem into a deterministic one with the assistance of the Lyapunov optimization approach in Section III-B.

III. Convergence Analysis and Problem Formulation

In this section, we theoretically analyze the convergence bound of the proposed KFL under a non-convex loss function setting. The convergence bound reveals that the scheduled data volume in each round and different learning rounds significantly affect the learning performance. Motivated by this, we define a new metric, i.e., the weighted scheduled data volume, to guide the device scheduling design. Then, we transfer the original problem to maximize this metric for minimizing the gap between the global loss function and the optimal loss. To enable the online dynamic device scheduling under long-term energy budgets constraint, we further transform the problem into a deterministic problem in each round with the assistance of the Lyapunov optimization approach.

A. Convergence Analysis

In this subsection, we investigate the convergence behavior of the proposed KFL algorithm. To facilitate the analysis, we make the following assumptions on each local loss function $F_k(\cdot)$.

Assumption 1. All empirical loss functions $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ ($k \in \mathcal{K}$) are continuously differentiable with respect to $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$, and there exist constants L_u, L_v, L_{uv}, and L_{vu} such that for each $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$:

- $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is L_u-Lipschitz continuous with $\boldsymbol{u}_k$ and L_{uv}-Lipschitz continuous with $\boldsymbol{v}_k$, that is,

$$\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}'_k, \boldsymbol{v}_k)\| \leq L_u \|\boldsymbol{u}_k - \boldsymbol{u}'_k\|, \tag{15}$$

and

$$\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}'_k)\| \leq L_{uv} \|\boldsymbol{v}_k - \boldsymbol{v}'_k\|. \tag{16}$$

- $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is L_v-Lipschitz continuous with $\boldsymbol{v}_k$ and L_{vu}-Lipschitz continuous with $\boldsymbol{u}$.

Assumption 2. The squared norm of gradients is uniformly bounded, i.e., $\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq$ G_1^2 and $\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq G_2^2$.

Assumption 3. For each local feature extractor $h_k(\cdot)$ ($\forall k \in \mathcal{K}$), its gradient norm is bounded by ϑ^2, i.e., $\|\nabla h_k(\boldsymbol{u}_k)\|^2 \leq \vartheta^2$, and the squared norm of its output vector is bounded by $\|h_k(\boldsymbol{u}_k; x)\|^2 \leq \varsigma^2$.

Assumption 1 is satisfied by most deep NNs. The modern NNs are usually composed of multiple layers. Based on [31], a deep NN defined by a composition of functions is a Lipschitz NN if the functions in all layers are Lipschitz. It has been proved in [31], [32] that the convolution layer, linear layer, and some nonlinear activation functions (e.g., Sigmoid and tanh) are Lipschitz functions. Thus, most deep NNs have Lipschitz continuous gradients. For a Lipschitz NN in which all layers are Lipschitz functions, both the feature extractor and predictor composed of Lipschitz layers are Lipschitz functions. Thus, Assumption 1 is satisfied by assuming the whole NN to be Lipschitz continuous. In addition, according to Proposition 1 in [31], one can derive that $F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ is $(L_u \times L_v)$-smooth based on Assumption 1. Assumption 2 is widely used in the existing convergence analysis works, e.g., [13]–[15], [17]. Assumption 3 is inherently satisfied by Assumption 2 since the gradient of a NN is a function of its output vector. To begin with, we first derive a key lemma to assist our analysis as follows:

Lemma 1. *Let Assumption* 1 *holds, we have*

$$F_k(\boldsymbol{u}_k', \boldsymbol{v}_k') - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \frac{1+\chi}{2} L_u \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|^2$$
$$+ \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}_k' - \boldsymbol{v}_k \rangle + \frac{1+\chi}{2} L_v \|\boldsymbol{v}_k' - \boldsymbol{v}_k\|^2, \quad (17)$$

where $\chi = \max\{L_{uv}, L_{vu}\}/\sqrt{L_u L_v}$, *which measures the relative cross-sensitivity of* $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ *with respect to* $\boldsymbol{v}_k$ *and* $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ *with respect to* $\boldsymbol{u}_k$.

Proof. Please see Appendix A. □

Lemma 1 reveals the gradient relationships of a NN between its feature extractor and label predictor part. According to Lemma 1, we derive the one-round convergence bound of any device k ($k \in \mathcal{K}$) in Lemma 2, in which devices utilize the proposed knowledge-aided loss to update their local models.

Lemma 2. *Let Assumption* 1, 2, *and* 3 *hold. The learning rates satisfy* $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ *and* $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$, *the one-round convergence bound of device* k ($k \in \mathcal{K}$) *is given by*

$$F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \leq \left(2(1+\chi)L_u \eta_u^2 \tau^2 - \frac{1}{2}\eta_u \tau\right) \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2$$
$$+ \left((1+\chi)L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau\right) \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + A_1 + \frac{5}{4}\eta_u \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$
$$+ 2\eta_u^2 \lambda^2 (3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v) \sum_{l=0}^{\tau-1} (\tau - l) \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2, \quad (18)$$

where $A_1 = \tau(\tau+1)(2\tau+1)\left(\eta_u^3 G_1^2 L_u^2 + \frac{1}{3}\eta_v^3 G_2^2 L_v^2 + (\frac{2}{3}\eta_u G_1^2 + \frac{1}{2}\eta_v G_2^2)\eta_u\eta_v\chi^2 L_u L_v\right)$.

Proof. Please see Appendix B. □

Based on Lemma 2, we further analyze the convergence behaviour of the proposed KFL algorithm after T rounds in Theorem 1, which takes into account the knowledge aggregation between devices.

Theorem 1. *Let Assumption* 1, 2, *and* 3 *hold,* $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ *and* $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$, *the gap between the global loss function after* T *rounds and the optimal loss is bounded by*

$$F(\boldsymbol{W}_T) - F(\boldsymbol{W}^*) \leq A_3^T(F(\boldsymbol{W}_0) - F(\boldsymbol{W}^*))$$
$$+ \frac{1-A_3^T}{1-A_3}(A_1+A_2) + A_2\frac{CK}{D}\sum_{t=0}^{T-2}A_3^{T-2-t}\sum_{k=1}^{K}\sum_{c=1}^{C}\frac{D_{k,c}^2}{D_c^2 D_k}\sum_{k=1}^{K}D_{k,c}^2$$
$$- A_2\frac{1}{DK(T-1)}\frac{1}{\max_{1\leq k\leq K}D_k}\left(\sum_{t=0}^{T-2}A_3^{T-2-t}\sum_{k=1}^{K}\alpha_{k,t}D_k\right)^2 \quad (19)$$

where $A_2 = 10\eta_u\lambda^2\tau\vartheta^2\varsigma^2 + 8\eta_u^2\lambda^2\vartheta^2\varsigma^2\left(3\eta_u L_u^2 + 2\eta_v\chi^2 L_u L_v\right)\tau(\tau+1)$, $A_3 = 1 + (4L_u^2\eta_u^2 + 2L_v^2\eta_v^2)(1+\chi)\tau^2 - (\eta_u L_u + \eta_v L_v)\tau$.

Proof. Please see Appendix C. □

Theorem 1 reveals how the device scheduling policy affects the convergence bound of KFL without characterizing the impact of non-IID degrees on the convergence bound. In general, the non-IID degree is characterized by the difference between the optimal global loss and the weighted summation of optimal local losses [33]. However, the proposed KFL is a personalized FL algorithm which trains a personalized model for each device. Thus, one cannot characterize the impacts of non-IID degree on the convergence bound in this way due to $F(\boldsymbol{W}^*) - \frac{1}{D}\sum_{k=1}^{K}D_k F_k(\boldsymbol{w}_k^*) = 0$. However, how to characterize non-IID degrees' effects on the convergence bound of personalized FL algorithms is a promising research direction, which will be studied in our future works.

According to Theorem 1, the gap between the global loss after T rounds and the optimal loss is bounded by four terms, 1) the gap in the initial round, 2) two terms related to hyperparameters of the learning system, 3) the scheduled data volume in all rounds. It is noted that $A_3 \leq 1$ due to $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ and $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$. As T increases, A_3^T approaches to 0. Hence, the first term converges to 0, and the second and the third terms converge to a constant. The first three terms decided by the system hyperparameters and initial models of devices are not related to the device scheduling policies. The last term is an explicit form related to device scheduling. For the last term, we have the following remark:

Remark 1. Increasing the scheduled data samples in each round is able to narrow the gap between global loss and optimal loss. In addition, as t increases, A_3^{T-1-t} also increases due to $A_3 < 1$. This indicates that more devices should be scheduled in early rounds when the total number of scheduled devices in the learning process is fixed.

Note that, it has been experimentally observed in [6] that scheduling more devices in the later rounds is beneficial for the learning performance of the federated averaging algorithm. However, the proposed KFL that only aggregates devices' knowledge in each round achieves better learning performance when scheduling more devices in the earlier rounds, which is verified by the theoretical analysis in Remark 1 and experimental results in Section V.

B. Problem Transformation via Lyapunov Optimization Framework

According to Theorem 1, the gap between the global loss and the optimal loss can be narrowed by minimizing the last term on the right-hand-side (RHS) of (19). However, it is tractable to directly minimize this term since it involves some unknown parameters, e.g., the Lipschitz constant L_u and L_v. Based on [32], the exact computation of the Lipschitz constant of deep learning architectures is intractable, even for two-layer NNs. Inspired by Remark 1, to enable tractable device scheduling design, we introduce a variable γ_t ($t = 0, 1, \cdots, T - 1$) as the weight of scheduled data samples in round t to capture the varying significance of scheduling devices in different rounds. Based on this, we define the weighted scheduled data volume as $\sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k$ and maximize it for the global loss minimization. Thus, we transform problem $\mathcal{P}$ as the following problem:

$$\widehat{\mathcal{P}} : \quad \max_{\{\boldsymbol{S}_t, \boldsymbol{\theta}_t, \boldsymbol{p}_t\}_{t=0}^{T-1}} \sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k \tag{20}$$

$$\text{s. t.} \quad (14a), (14b), (14c), (14d), (14e).$$

Problem $\widehat{\mathcal{P}}$ involves multi-dimension discrete and continuous variables is a typical mixed-integer programming problem, which is generally NP-Hard. In addition, solving the optimal solution of problem $\widehat{\mathcal{P}}$ offline requires optimally dividing the energy of all devices in each round due to the long-term energy constraints, which is intractable. The most critical challenge of directly solving problem $\widehat{\mathcal{P}}$ is that it requires channel information of all devices over all rounds at the beginning of the FL process, which may unfeasible in practical systems. To enable the online dynamic device scheduling, we utilize the Lyapunov optimization framework to deal with the correlations among rounds. To this end, we construct a virtual queue $q_k(t)$ for each device k ($k \in \mathcal{K}$), which evolves as

$$q_k(t+1) = \max \left\{ q_k(t) + \alpha_{k,t} E_{k,t} - \frac{E_k}{T}, 0 \right\}, \tag{21}$$

with the initial value $q_k(t) = 0$ for all devices. Inspired by the drift-plus-penalty algorithm in [34], we transform problem $\widehat{\mathcal{P}}$ as the following problem to enable online device scheduling

$$\widetilde{\mathcal{P}}: \quad \min_{\{ \boldsymbol{S}_t, \boldsymbol{\theta}_t, \boldsymbol{p}_t \}_{t=0}^{T-1}} - V \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_k + \sum_{k=1}^{K} q_k(t) \alpha_{k,t} E_{k,t} \tag{22}$$

$$\text{s. t.} \quad (14\text{b}), (14\text{c}), (14\text{d}), (14\text{e}).$$

In problem $\widetilde{\mathcal{P}}$, $V \geq 0$ is a weight factor that balances the energy consumption of devices and learning performance. A large V emphasises the learning performance improvement by sacrificing the devices' energy and vice versa. In addition, from the objective function (22), the unscheduled devices in the former rounds have smaller $q_k(t)$. These devices are encouraged to participate in the current round of training for minimizing (22). Thus, problem $\widetilde{\mathcal{P}}$ contributes to a fair scheduling scheme between devices.

IV. ONLINE DEVICE SCHEDULING AND WIRELESS RESOURCE ALLOCATION

In this section, we propose an energy-aware device scheduling, bandwidth allocation, and power control algorithm that solves problem $\widetilde{\mathcal{P}}$ in an online fashion. We first derive the optimal bandwidth allocation and power control policies using convex optimization techniques. Then, we propose a polynomial-time algorithm to solve the device scheduling decision with a $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off guarantee, where V is an algorithm-related parameter.

A. Optimal Power Control and Bandwidth Allocation

For any given scheduled device set $\boldsymbol{S}_t \in \mathcal{K}$, we decompose the bandwidth allocation and power control problem from $\widetilde{\mathcal{P}}$ as follows:

$$\mathcal{P}_1: \quad \min_{\{ \boldsymbol{\theta}_t, \boldsymbol{p}_t \}} \sum_{k \in \boldsymbol{S}_t} q_k(t) E_{k,t} \tag{23}$$

$$\text{s. t.} \quad (14\text{b}), (14\text{c}), (14\text{d}).$$

For problem $\mathcal{P}_1$, we have the following proposition:

Proposition 1. *The optimal solution of problem $\mathcal{P}_1$ satisfies $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}$, and the optimal transmit power of device k satisfies*

$$p_{k,t} = \frac{\theta_{k,t} B N_0}{h_{k,t}} \left(2^{\frac{Qq}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}) \theta_{k,t} B}} - 1 \right). \tag{24}$$

Sketch of proof: By proving that the first-order derivatives of the objective function (23) are great than 0, (23) is an non-increasing function with respect to the communication time $\mathcal{T}_{k,t}^{\mathrm{U}}$.

Thus, the optimal completion time of device k is $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}$. Based on (13), the proposition is proved. We have the detailed proof in Section I of the technical report [35].

According to Proposition 1, we substitute (24) into problem $\mathcal{P}_1$, the optimal bandwidth allocation problem can be formulated as

$$\mathcal{P}_2: \quad \min_{\boldsymbol{\theta}_t} \ \sum_{k \in \boldsymbol{S}_t} \frac{\theta_{k,t} B N_0 q_k(t)(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}{h_{k,t}} \mathcal{I}(\theta_{k,t}) \tag{25}$$

$$\text{s. t. } (14c), (14d),$$

$$\theta_{k,t} B \log \left(1 + \frac{p_{k,\max} h_{k,t}}{\theta_{k,t} B N_0} \right) \geq \frac{Qq}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})}, \tag{25a}$$

where

$$\mathcal{I}(\theta_{k,t}) = \exp \left(\frac{Qq \ln 2}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})\theta_{k,t} B} \right) - 1. \tag{26}$$

For problem $\mathcal{P}_2$, we obtain its optimal solution by using the following lemma.

Lemma 3. *The optimal bandwidth allocation of problem $\mathcal{P}_2$ satisfies*

$$\theta_{k,t}^* = \max \left\{ \theta_{k,t}(\mu), \theta_{k,t}^{\min} \right\}, \tag{27}$$

where
$$\theta_{k,t}(\mu) = \frac{Qq \ln 2}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}) B \left(\mathcal{W} \left(\frac{\mu h_{k,t}}{eB N_0 q_k(t)(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})} - \frac{1}{e} \right) + 1 \right)}, \tag{28}$$

and $\theta_{k,t}^{\min}$ satisfies constraint (14c), μ is the Lagrange multiplier which satisfies $\sum_{k=1}^K \theta_{k,t}(\mu^*) = 1$. $\mathcal{W}(\cdot)$ is the principal branch of the Lambert function, defined as the solution of $\mathcal{W}(x)e^{\mathcal{W}(x)} = x$, in which e is the Euler's number.

Sketch of proof: The problem $\mathcal{P}_2$ is a convex problem. By solving the KKT conditions of $\mathcal{P}_2$, the lemma is proved. We have the detailed proof in Section II of the technical report [35].

Although Lemma 3 provides the optimal condition of bandwidth allocation, there is still an unknown variable μ. Below we develop a binary search method to solve the optimal μ. Since the Lagrange multiplier $\mu \geq 0$, we have $\frac{\mu h_{k,t}}{eB N_0 q_k(t)(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})} - \frac{1}{e} \geq -\frac{1}{e}$. Moreover, $\mathcal{W}(x)$ is a monotonically increasing function when $x \geq -\frac{1}{e}$. Thus, $\theta_{k,t}(\mu)$ is a monotonically decreasing function with respect to μ. To deploy the binary search method, we derive the lower and upper bound of μ. Since $\mu \geq 0$, the lower bound of μ is $\mu_{\mathrm{lb}} = 0$. For the upper bound, we have $\max_{\boldsymbol{S}_t} \{\theta_{k,t}(\mu)\} \geq \frac{1}{|\boldsymbol{S}_t|}$. Let $\varphi_k = \frac{Qq|\boldsymbol{S}_t| \ln 2}{(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}})B}$. Based on the definition of Lambert function, we have

$$\mu \leq \mu_{\mathrm{ub}} = \max_{k \in \boldsymbol{S}_t} \left\{ \frac{B N_0 q_k(t)(\mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}) \left((\varphi_k - 1)e^{\varphi_k} + 1 \right)}{h_{k,t}} \right\}. \tag{29}$$

Based on the lower bound μ_{lb} and upper bound μ_{ub}, the optimal Lagrange multiplier can be obtained by the binary search method. For clarity, we summarize the detailed steps for solving the optimal bandwidth allocation policy in Algorithm 2. The binary search method halves the

search region at every iteration and terminate when the given precision (i.e., ε) requirement is satisfied. Thus, the time complexity of this method is $\mathcal{O}\left(\log_2 \frac{\mu_{\mathrm{ub}} - \mu_{\mathrm{lb}}}{\varepsilon}\right)$.

Algorithm 2 Optimal Wireless Bandwidth Allocation

1: Initialize $\boldsymbol{S}_t$, the precision requirement $\varepsilon > 0$.
2: Initialize the upper bound of Lagrange multiplier μ_{ub} based on (29), set the lower bound to $\mu_{\mathrm{lb}} = 0$.
3: **repeat**
4:      Set $\mu = (\mu_{\mathrm{lb}} + \mu_{\mathrm{ub}})/2$.
5:      For each device $k \in \boldsymbol{S}_t$, compute the required bandwidth allocation ratio $\theta_{k,t}(\mu)$ based on (28).
6:      Compute the summation of required bandwidth allocation ratio $\sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mu)$.
7:      **if** $\sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mu) > 1$ **then**
8:          Halve the searching region by setting $\mu_{\mathrm{lb}} = \mu$ and $\mu_{\mathrm{ub}} = \mu_{\mathrm{ub}}$.
9:      **else if** $0 < \sum_{k \in \boldsymbol{S}_t} \theta_{k,t}(\mu) < 1 - \varepsilon$ **then**
10:          Halve the searching region by setting $\mu_{\mathrm{lb}} = \mu_{\mathrm{lb}}$ and $\mu_{\mathrm{ub}} = \mu$.
11:      **else**
12:          Break the circulation.
13:      **end if**
14: **until** $|\mu_{\mathrm{ub}} - \mu_{\mathrm{lb}}| < \varepsilon$
15: Substituting μ into (28) for get $\theta_{k,t}(\mu)$, then compute the optimal bandwidth allocation policy based on (27).
16: Substitute the optimal bandwidth allocation policy into (24) for obtaining the optimal power control policy.
17: **return** The optimal bandwidth allocation policy $\boldsymbol{\theta}_t$, the optimal power control policy $\boldsymbol{p}_t$.

B. Device Scheduling

Based on the above analysis, the optimal bandwidth allocation and power control policy for any device scheduling set $\boldsymbol{S}_t$ can be obtained by using Algorithm 2. For device scheduling design, an intuitive method is to compute the objective function value for all possible device scheduling decisions, and select the one with minimal objective function as the final scheduling decision. However, this intuitive method is infeasible in its implementation since there are $\sum_{n=0}^{K} C_K^n = 2^K$ possible scheduling decisions, inducing an exponential time complexity with $\mathcal{O}\left(2^K \log_2 \frac{\mu_{\mathrm{ub}} - \mu_{\mathrm{lb}}}{\varepsilon}\right)$. In the following part, we develop an efficient algorithm to solve the device scheduling policy.

According to the objective function (22), it is desirable to select devices with small $q_k(t)$ and $E_{k,t}$, as well as large data samples. The small $E_{k,t}$ is achieved by strong channels and low computation energy consumption. To identify these devices, we first allocate equal bandwidth to all devices (i.e., $\theta = 1/K$), and then substitute $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}$ into (13) to compute the estimated energy consumption $\bar{E}_{k,t} = E_{k,t}^{\mathrm{L}} + E_{k,t}^{\mathrm{U}}$. Based on the estimated energy consumption for all devices, we sort devices based on $\Delta_{k,t} = -V\gamma_t D_{k,c} + q_k(t)\bar{E}_{k,t}$ ($\forall k \in \mathcal{K}$) in the ascending order. Denote $\widetilde{\mathcal{K}}$ as the sorted device set. Many sorting algorithms, such as Heapsort or Mergesort, can be used, with a worst-case complexity $\mathcal{O}(K \log K)$. Then, we solve the device scheduling policy by incrementally adding devices into the selection set $\boldsymbol{S}$ from the sorted device set $\widetilde{\mathcal{K}}$. For each possible device scheduling set $\boldsymbol{S}$, we perform Algorithm 2 to obtain the optimal wireless bandwidth allocation $\boldsymbol{\theta}_t^*(\boldsymbol{S})$, power control decisions $\boldsymbol{p}_t^*(\boldsymbol{S})$, as well as the optimal energy consumption $E_t^*(\boldsymbol{S})$. Substituting $E_t^*(\boldsymbol{S})$ into (22), the drift-plus-penalty value of device scheduling set $\boldsymbol{S}$ can be obtained, denoted as $\mathcal{Y}(\boldsymbol{S})$. Let $\mathcal{H}$ denote the set of all possible device

scheduling set S. Finally, we obtain the optimal device scheduling policy through comparing the drift-plus-penalty value of all possible device scheduling set $S \in \mathcal{H}$, i.e., $S_t^* = \arg\min_{S \in \mathcal{H}} \mathcal{Y}(S)$. Note that, the energy consumption of devices with $q_k(t) = 0$ does not affect the objective function value, the minimal required bandwidth should be allocated to them for saving more bandwidth for other users with $q_k(t) > 0$. For clarity, we summarize the detail steps of device scheduling algorithm in Algorithm 3, which obtains the device scheduling policy by performing at most K times Algorithm 2 and has a polynomial time complexity $\mathcal{O}\left(K \log_2 \frac{\mu_{\mathrm{ub}} - \mu_{\mathrm{lb}}}{\varepsilon}\right)$.

For Algorithm 3, we analyze its performance by comparing it with its optimal offline counterpart which is in fact problem $\widehat{\mathcal{P}}$. The offline algorithm has the channel information of all rounds. Let $\alpha_{k,t}^*$ be the offline optimal device scheduling decision obtained by solving the problem $\widehat{\mathcal{P}}$ with pre-known device information. The performance guarantee of the proposed device scheduling algorithm is shown in Proposition 2.

Algorithm 3 Energy-aware online Device scheduling

1: Input the virtual queue length $q_k(t)$ ($k \in \mathcal{K}$) and γ_t, initialize V.
2: Substituting $\theta_{k,t} = 1/K$ and $\mathcal{T}_{k,t}^{\mathrm{U}} = \mathcal{T}_{\max} - \mathcal{T}_k^{\mathrm{L}}$ into (13) to compute the estimated energy consumption of device k ($\forall k \in \mathcal{K}$), i.e.,
   $\bar{E}_{k,t} = E_{k,t}^{\mathrm{L}} + E_{k,t}^{\mathrm{U}}$.
3: Sort devices based on $\Delta_{k,t} = -V\gamma_t D_{k,c} + q_k(t)\bar{E}_{k,t}$ in the ascending order to obtain the sorted device set $\widetilde{\mathcal{K}}$.
4: **for** $k = \widetilde{\mathcal{K}}(1), \widetilde{\mathcal{K}}(2), \cdots, \widetilde{\mathcal{K}}(K)$ **do**
5:    Update $S = S \cup \{k\}$
6:    Solve the optimal bandwidth allocation and power control policy by Algorithm 2, i.e., $\theta_t^*(S)$ and $p_t^*(S)$.
7:    Compute the drift-plus-penalty of S, i.e., $\mathcal{Y}(S) = -V\gamma_t \sum_{k \in S} D_k + \sum_{k \in S} q_k(t)E_{k,t}$
8:    **if** $-VD_k + q_k(t)E_{k,t} > 0$ **then**
9:       Break the circulation
10:   **else**
11:       Add S into $\mathcal{H}$, i.e., $\mathcal{H} = \mathcal{H} \cup S$
12:   **end if**
13: **end for**
14: Find the optimal device scheduling set $S_t^* = \arg\min_{S \in \mathcal{H}} \mathcal{Y}(S)$.
15: **return** The device scheduling set S_t^*, wireless bandwidth allocation $\theta_t^*(S_t^*)$, and power control policy $p_t^*(S_t^*)$.

Proposition 2. *Compared to the offline optimal solution, the cumulative loss of Algorithm 3 is bounded by*

$$\sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t} D_{k,c} \geq -\frac{T\zeta_0}{V} - \frac{T(T-1)}{2V} \sum_{k=1}^{K} \zeta_k^2 + \sum_{t=0}^{T-1} \gamma_t \sum_{k=1}^{K} \alpha_{k,t}^* D_{k,c}, \quad (30)$$

and the total energy consumption of Algorithm 3 is bounded by

$$\sum_{t=0}^{T-1} \sum_{k=1}^{K} \alpha_{k,t} E_{k,t} \leq \sum_{k=1}^{K} E_k + \sqrt{2K\left(T\zeta_0 + V \sum_{t=0}^{T-1} \gamma_t D\right)}, \quad (31)$$

where $\zeta_0 = \frac{1}{2} \sum_{k=1}^{K} \zeta_k^2$ and $\zeta_k = \max_t \left\{\left|\alpha_{k,t} E_{k,t} - \frac{E_k}{T}\right|\right\}$.

Proof. Please see Appendix D □

Proposition 2 characterizes the performance of the proposed device scheduling algorithm, which shows that 1) the energy constraints of devices are approximately satisfied with the $\mathcal{O}(\sqrt{V})$-bounded factor, and 2) the proposed device algorithm is $\mathcal{O}(1/V)$-optimal with respect to

the performance of its optimal offline counterpart solution. Thus, the proposed device scheduling algorithm demonstrates an $\mathcal{O}(\sqrt{V}, 1/V)$ energy-learning trade-off. The worst-case performance of Algorithm 3 can be improved by reducing the upper bound of energy usage bias ζ_0. In addition, adjusting the weight parameter V is able to achieve the balance between the learning performance and energy consumption of devices. Specifically, with larger V, more emphasis is put on the scheduled data samples to improve the learning performance while more energy is consumed at devices, and vice versa. In practical systems, one should carefully select the value of V to optimize the learning performance with energy limits and use the energy in a balanced manner to avoid large ζ_k.

V. NUMERICAL RESULTS

In this section, we verify the effectiveness of the proposed KFL algorithm. If not specified, we consider $K = 100$ devices randomly distributed in a cell with a radius of 500m, and the server is located at the centre of the cell. The total bandwidth is set to $B = 5$MHz. Similar to [36], the channel gain is modelled as $h_{k,t} = h_0 \rho_k(t)(d_0/d_k)^v$, where $h_0 = -30$dBm is the path loss constant; d_k is the distance between device k and the edge server; $d_0 = 1$m is the reference distance; $\rho_k(t) \sim \text{Exp}(1)$ is exponentially distributed with unit mean, which represents the small-scale fading channel power gain from the device k to the edge server in round t; d_0/d_k represents the large-scale path loss with $v = 2$ being the path loss exponent. The channel noise power spectral density N_0 is set to -174 dBm. For all devices in the system, we set their maximal transmit power to $p_{k,\max} = 30$dBm, and their CPU frequency are randomly selected from $\{0.85, 1.12, 1.2, 1.3\}$GHz [37], [38].

We evaluate the proposed KFL algorithm on two image classification tasks using MNIST and CIFAR-10 datasets, both of them have 10 classes of data samples. For the MNIST dataset, we train five-layer multilayer perceptron (MLP) models with the following architecture: four fully connected layers with 784, 512, d_{MLP}, 64 units, each of these layers is activated by the ReLU function; and a 10-unit softmax output layer. For the CIFAR-10 dataset, we train five-layer CNN models with the following structure: two $5{\times}5$ convolution layers followed by a $2{\times}2$ max-pooling layer, in which the first convolution layer possesses 6 channels and the second layer with 16 channels; three fully connected layers with 400, d_{CNN}, and 64 units, respectively; and a 10-unit softmax output layer. The ReLU function activates each convolution or fully connected layer. When devices are equipped with homogeneous models, we set $d_{\text{MLP}} = 256$ and $d_{\text{CNN}} = 128$. The number of FLOPs and parameters of these machine learning models can be estimated

using the method in [27]. Specifically, the MLP with $d_{\text{MLP}} = 256$ possesses 553406 parameters which equal the FLOPs required to process one data sample. The CNN with $d_{\text{CNN}} = 128$ has 63106 parameters and requires 1245834 FLOPs to process one data sample. When devices have heterogeneous models, the value of d_{MLP} and d_{CNN} for all devices are randomly selected from $\{128, 192, 256, 320, 384\}$. For both MLP and CNN, the learning rates, i.e., η_u and η_v, are set to 0.05, a momentum of 0.9 is adopted, and the number of local iterations is set to $\tau = 5$, and cross-entropy is adopted as the loss function. In addition, we first classify the training data samples according to their labels, then split each class of data samples into $mK/10$ shards, and finally randomly distribute two shards of data samples to each device. That is, each client has a data distribution corresponding to at most m classes. The data distributions among devices are more skewed for smaller m. Due to page limits, we only show results based on $m = 2$ in the following experiments. The results based on $m = 3$ are presented in Section III of our technical report [35], showing similar results with $m = 2$. For the MNIST dataset, we set the energy budgets $E_k = 0.1 \times T$ J $(\forall k \in \mathcal{K})$ and $T_{\max} = 1$s. For CIFAR-10, we set $E_k = 0.5 \times T$ J $(\forall k \in \mathcal{K})$ and $T_{\max} = 2$s. In the simulations, each device first computes the number of correct predicted data samples on its test dataset by its local model. Note that the deployed trained models on devices are the same in FedAvg, while each device has a personalized local model in the proposed KFL. Then, the test accuracy is computed as the total number of correct predicted test data samples on all devices divided by the total number of test data samples on all devices.

In the following sections, we verify the theoretical results in Remark 1 on MNIST and CIFAR-10 datasets by comparing the learning performance of the following three temporal device scheduling patterns. 1) Uniform Scheduling: Ten devices are randomly scheduled in each round to participate in the learning process. 2) Ascend Scheduling: The number of scheduled devices increases from 1 to 20, with an average number of 10 devices scheduled in each round. 3) Descend Scheduling: The number of scheduled devices decreases from 20 to 1, with an average number of 10 devices per round.

A. Performance Evaluation with Homogeneous Models

We evaluate the performance of the proposed KFL algorithm by comparing it with the following benchmarks. Note that, devices are equipped with homogeneous local models, and we do not consider the energy and bandwidth limitation in this subsection. 1) FedAvg [28]: In each round, the scheduled devices upload their model parameters to the edge server for aggregation. 2) FedRep [39]: The scheduled devices sequentially train the feature extractor and label predictor
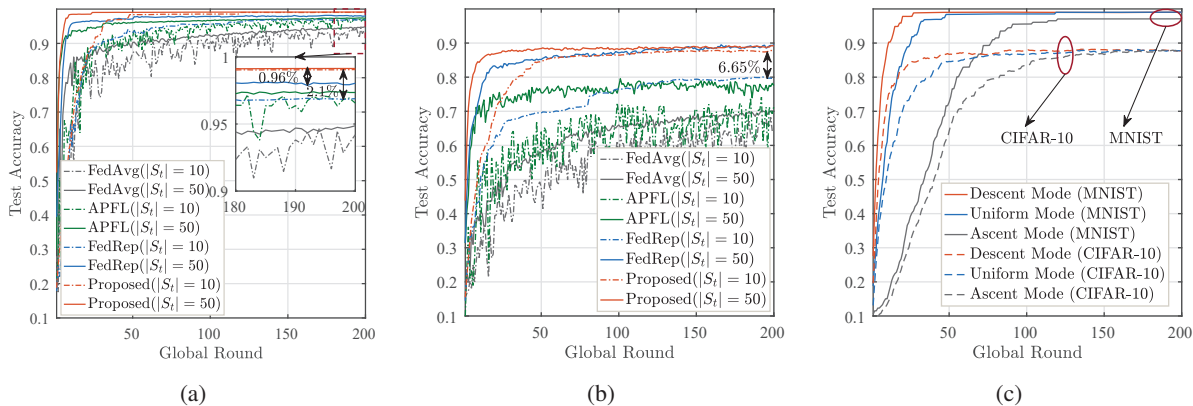
Fig. 2. Comparison of learning performance under homogeneous models (a) different algorithms on the MNIST dataset; (b) different algorithms on the CIFAR-10 dataset; (c) different scheduling patterns on MNIST and CIFAR-10 datasets.

parts of their models in each round. Particularly, the scheduled devices only upload feature extractor part of their models to the edge server for aggregation. 3) APFL [40]: In each round, each scheduled device trains its own local model and the received global model from the edge server. Then APFL incorporates the devices' locally trained model and the updated global model to achieve a device-specific model. It is worth mentioning that the proposed KFL algorithm requires fewer parameters transmission in each round than the benchmarks and thus reduces the communication cost. Specifically, for the MNIST dataset, the proposed algorithm only requires devices to upload the knowledge of 10 classes, including $64 \times 10 = 640$ parameters, accounting for 0.12% of the transmitted parameters by FedAvg or by FedRep. For the CIFAR-10 dataset, the KFL algorithm only requires devices to upload 640 parameters in each round, comprising 1.01% of the total model parameters.

Fig. 2 shows the learning performance of the proposed KFL algorithm and two benchmarks on MNIST and CIFAR-10 datasets. From Fig. 2(a), compared to the state-of-art FedRep, it is observed that the proposed algorithm achieves 0.96% accuracy improvement when 50 devices participate in each round learning process and obtains a 2.1% accuracy gain when scheduling 10 devices in each round. In addition, the proposed algorithm converges faster than the benchmarks. A similar experiment conducted on the CIFAR-10 dataset is shown in Fig. 2(b). Similar to the results on the MINIST dataset, the proposed algorithm outperforms the benchmarks. Specifically, it improves 6.65% accuracy when 10 devices are scheduled in each round. Although the proposed algorithm has similar accuracy to the FedRep when scheduling 50 devices in each round, it converges faster than the latter.

Fig. 2(c) presents the test accuracy of the proposed KFL algorithm with different device

scheduling patterns on MNIST and CIFAR-10 datasets. It is observed that the descend scheduling pattern converges faster than the other two scheduling patterns on these two datasets. This experimental result verifies the theoretical results in Remark 1, which indicates that more scheduled data volume should bias to the early rounds if the entire scheduled data volume are fixed.

B. Performance Evaluation with Heterogeneous Models

In this subsection, we verify the effectiveness of the proposed KFL algorithm by comparing it with the FedKD [41], which is a knowledge distillation-based FL algorithm. Note that in this subsection, devices are equipped with heterogeneous local models, and do not consider the energy and bandwidth limitations. Since the knowledge distillation process requires aggregating devices' model output logits on an additional proxy dataset, we sample 50 data samples from each class (for both MNIST and CIFAR-10) to construct the proxy dataset with 500 data samples. Note that as stated in the experimental setting, our proposed algorithm only requires devices to upload 640 parameters in each round, reducing 87% of transmission costs compared with the knowledge distillation-based algorithm because the latter requires devices to upload $500 \times 10 = 5000$ parameters in each round.

Fig. 3 presents the test accuracy of the proposed KFL algorithm and the knowledge distillation-based FL algorithm under heterogeneous devices' models. Fig. 3(a) shows the results of the MNIST dataset. Compared to the FedKD algorithm, the proposed KFL algorithm achieves a slight test accuracy improvement, i.e., 0.61% when 10 devices and 0.59% when 50 devices participate in the per-round training. In addition, the proposed KFL algorithm convergences faster than the FedKD algorithm. Fig. 3(b) presents the results of the CIFAR-10 dataset which is more complex than the MNIST dataset. The proposed KFL algorithm obtains a more distinct learning performance gain on the CIFAR-10 dataset, i.e., compared to FedKD, improving 4% and 9.35% accuracy when 10 and 50 devices participate in per-round training, respectively. In fact, the learning performance of FedKD or other knowledge distillation-based FL algorithms heavily relies on the quality of the proxy dataset. In practical applications, the additional proxy dataset may not always be available, and its quality is usually not very high. Thus, the proposed KFL algorithm is flexible for practical scenarios. Fig. 3(c) shows a similar result to the experiment under homogeneous models, indicating that more data sample volume should be scheduled in the earlier rounds when the total scheduled data volume in the entire learning course are fixed.
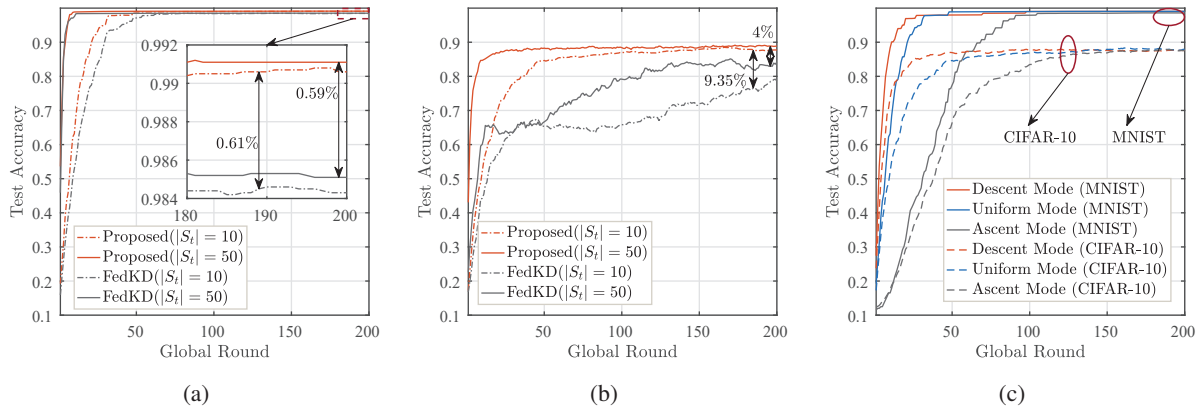
Fig. 3. Comparison of learning performance under heterogeneous models (a) different algorithms on the MNIST dataset; (b) different algorithms on the CIFAR-10 dataset; (c) different scheduling patterns on MNIST and CIFAR-10 datasets.
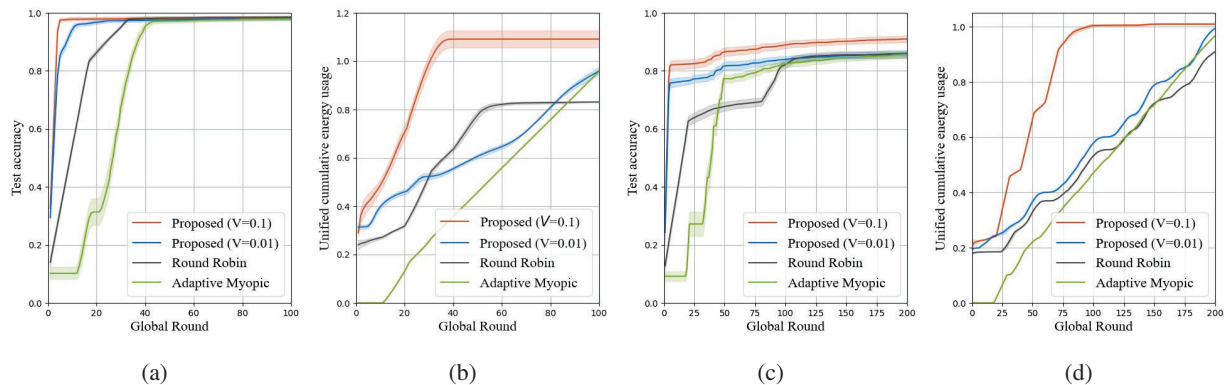

Fig. 4. Comparison of learning performance in different device scheduling algorithms on the MNIST and CIFAR-10 datasets.

C. Performance of the Proposed Device Scheduling Algorithm

This subsection evaluates the proposed device scheduling algorithm in the wireless network by comparing it with two benchmarks. Note that devices in this subsection are equipped with heterogeneous models. 1) Round Robin Scheduling Policy [42]: In each round, the round robin policy selects a set of devices with the size of 5 (for both MNIST and CIFAR-10 dataset) that have sufficient energy to support its current local training and knowledge uploading to participate in the training process. This policy contributes a fairness scheduling among devices. The size of the scheduled device set of the round robin is determined by the maximum overall scheduled devices of other scheduling algorithms divided by the number of rounds. 2) Myopic Scheduling Policy [7]: For each device k, the available energy in round t is given by the remaining energy divided by the remaining number of rounds, i.e., $\frac{E_k - \sum_{t'=0}^{t-1} \alpha_{k,t'} E_{k,t'}}{T-t+1}$. Note that, in this subsection, devices are equipped with heterogeneous models. For the proposed algorithm, we set $\gamma_t = \frac{1}{t}$ ($\forall t \in \{0, 1, \cdots, T-1\}$). Fig. 4(a) and Fig. 4(b) compare the test accuracy and cumulative energy usage of the scheduling algorithms on the MNIST dataset. It is observed from Fig. 4(a) that

the proposed algorithm obtains a faster convergence speed and higher test accuracy than the benchmarks. From Fig. 4(b), we can see that the proposed scheduling algorithm with $V = 0.01$ and $V = 0.1$ have higher energy usage in the beginning 30 rounds than benchmarks. This induces a faster convergence speed of the proposed algorithm. Particularly, the proposed algorithm with $V = 0.01$ has the same energy usage as the Adaptive Myopic algorithm. Both satisfy the energy constraints of devices (at the end of the training process, the unified energy usage is smaller than 1). However, the proposed algorithm with $V = 0.01$ achieves better learning performance. This performance gain comes from the proposed algorithm enabling devices to use energy more flexibly, thus improving the training performance.

Similar comparison is made on CIFAR-10 dataset in Fig. 4(c) and Fig. 4(d). It is also observed that the proposed online device scheduling algorithm outperforms the baselines in accuracy and convergence speed. From Fig. 4(d), we can see that the proposed algorithm enables devices to consume more energy in the earlier rounds compared to the baselines, which indicates that the proposed algorithm schedules more data samples in the early rounds. Thus, based on Remark 1, the proposed algorithm obtains better learning performance than the baselines. Particularly, the proposed algorithm with $V = 0.1$ enables devices to exhaust their energy in the former 100 rounds and achieve the best learning performance. The round robin algorithm enables devices to consume energy uniformly throughout the process. While for the Adaptive Myopic algorithm, the energy consumption at the former rounds exceeds the budget, and thus no devices are scheduled. In fact, the proposed algorithm schedules devices in the descend scheduling pattern, while Adaptive Myopic schedules devices in the ascend scheduling pattern and Round Robin schedules devices in the uniform scheduling pattern. Thus, the result in Fig. 4(a) and 4(d) also verified the correctness of our theoretical analysis in Remark 1, i.e., more data samples should be scheduled in the early rounds under restricted resources budgets.

VI. CONCLUSION

In this work, we have developed a novel KFL framework which aggregates devices' knowledge to enable collaborative training between devices. The benefits of this framework are three folds: 1) Allowing devices with heterogeneous models to train machine learning models collaboratively. 2) Significantly reducing the communication overhead of devices compared to conventional model aggregation-based FL approaches. 3) Mitigating the impact of non-IID data distribution among devices on learning performance. Experimental results show that compared to conventional model aggregation-based FL algorithms, the proposed KFL framework is able to reduce

99% communication load while boosting 2.1% and 6.65% accuracy on MNIST and CIFAR-10 datasets, respectively. In addition, we have theoretically and experimentally revealed that more scheduled data samples should be biased to the early rounds if the scheduled data samples of the entire learning process are fixed. With this insight, we have developed an efficient online device scheduling and resource allocation algorithm to improve learning performance under devices' limited energy budgets. Experimental results show that the proposed online device scheduling algorithm converges faster than the benchmark device scheduling algorithms. In the future work, we will optimize the local models' design according to the devices' computing capabilities and datasets for further improving the learning performance of KFL.

APPENDIX

A. Proof of Lemma 1

Using L_u smooth of $F_k(\cdot, \boldsymbol{v}_k)$ and L_v-smooth of $F(\boldsymbol{u}_k, \cdot)$, we have

$$F_k(\boldsymbol{u}_k', \boldsymbol{v}_k') - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k') \leq \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k'), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \frac{L_u}{2} \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|^2, \quad (32)$$

and

$$F_k(\boldsymbol{u}_k, \boldsymbol{v}_k') - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}_k' - \boldsymbol{v}_k \rangle + \frac{L_v}{2} \|\boldsymbol{v}_k' - \boldsymbol{v}_k\|^2. \quad (33)$$

Summarizing (32) and (33), we have

$$F_k(\boldsymbol{u}_k', \boldsymbol{v}_k') - F_k(\boldsymbol{u}_k, \boldsymbol{v}_k) \leq \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k'), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \frac{L_u}{2} \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|^2$$
$$+ \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{v}_k' - \boldsymbol{v}_k \rangle + \frac{L_v}{2} \|\boldsymbol{v}_k' - \boldsymbol{v}_k\|^2. \quad (34)$$

We now focus on bounding $\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k'), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle$ as follows:

$$\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k'), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle$$
$$\overset{(a)}{=} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k') - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle$$
$$\overset{(b)}{\leq} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k') - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)\| \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|$$
$$\overset{(c)}{\leq} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + L_{uv} \|\boldsymbol{v}_k' - \boldsymbol{v}_k\| \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|$$
$$\overset{(c)}{\leq} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k), \boldsymbol{u}_k' - \boldsymbol{u}_k \rangle + \frac{1}{2}\chi L_v \|\boldsymbol{v}_k' - \boldsymbol{v}_k\|^2 + \frac{1}{2}\chi L_u \|\boldsymbol{u}_k' - \boldsymbol{u}_k\|^2, \quad (35)$$

where (a) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k)$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_k, \boldsymbol{v}_k')$, (b) follows the Cauchy-Schwarz inequality, (c) comes from Assumption 1, (d) is due to the definition of χ. Substituting (35) into (34), the proof completes.

B. Proof of Lemma 2

According to Lemma 1, we have

$$F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \leq \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle + \frac{1+\chi}{2} L_u \|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2$$
$$+ \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t} \rangle + \frac{1+\chi}{2} L_v \|\boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t}\|^2. \quad (36)$$

Below we focus on bounding the four terms on the right-hand side (RHS) of (36). Firstly, we bound $\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle$ as follows:

$$\langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t} \rangle = -\eta_u \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l}) \rangle$$

$$\overset{(a)}{\leq} -\frac{\eta_u \tau}{2} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + \frac{\eta_u}{2} \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$

$$\overset{(b)}{\leq} -\frac{\eta_u \tau}{2} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + \eta_u \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$

$$+ \eta_u \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \quad (37)$$

where (a) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality, (b) follows the triangle inequality. For the second term on the RHS of (36), we bound $\|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2$ as

$$\|\boldsymbol{u}_{k,t+1} - \boldsymbol{u}_{k,t}\|^2 = \eta_u^2 \left\| \sum_{l=0}^{\tau-1} (\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})) \right\|^2$$

$$\overset{(a)}{\leq} \eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$

$$\overset{(b)}{\leq} 2\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})\|^2 + 2\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\lambda \nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$

$$\overset{(c)}{\leq} 4\eta_u^2 \tau^2 \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 + 2\eta_u^2 \tau \lambda^2 \sum_{l=0}^{\tau-1} \|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$$

$$+ 4\eta_u^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \quad (38)$$

where (a) is due to Jensen's inequality, (b) follows the triangle inequality, (c) is derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$. We now focus on bounding $\sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2$ which appears in both (37) and (38) as

$$\sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2$$

$$\overset{(a)}{\leq} 2 \sum_{l=0}^{\tau-1} \left(\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l})\|^2 + \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \right)$$

$$\overset{(b)}{\leq} 2 \sum_{l=0}^{\tau-1} L_u^2 \|\boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t}\|^2 + 2 \sum_{l=0}^{\tau-1} \chi^2 L_u L_v \|\boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t}\|^2. \quad (39)$$

where (a) derived by adding and substracting $\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality, (b) follows Assumption 1 and the definition of χ.

For the last two terms on the RHS of (36), we have

$$\langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t} \rangle + \frac{1+\chi}{2} L_v \|\boldsymbol{v}_{k,t+1} - \boldsymbol{v}_{k,t}\|^2$$

$$= -\eta_v \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \rangle + \frac{1+\chi}{2} L_v \eta_v^2 \left\| \sum_{l=0}^{\tau-1} \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \right\|^2$$

$$\overset{(a)}{\leq} -\eta_v \sum_{l=0}^{\tau-1} \langle \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}), \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \rangle + \frac{1+\chi}{2} L_v \eta_v^2 \tau \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})\|^2$$

$$\overset{(b)}{\leq} ((1+\chi) L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau) \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2$$

$$+ ((1+\chi) L_v \eta_v^2 \tau + \frac{1}{2}\eta_v) \sum_{l=0}^{\tau-1} \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2, \quad (40)$$

where (a) is due to Jensen's inequality, (b) is derived by adding and substracting $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})$ into $\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l})$ and using the triangle inequality. In (40), we bound $\sum_{l=0}^{\tau-1} \| -\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) + \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) \|^2$ as

$$
\sum_{l=0}^{\tau-1} \| \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2
$$

$$
\leq 2 \sum_{l=0}^{\tau-1} \| \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,l}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l}) \|^2 + 2 \sum_{l=0}^{\tau-1} \| \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t,l}) - \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2
$$

$$
\leq 2 \sum_{l=0}^{\tau-1} \chi^2 L_u L_v \| \boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t} \|^2 + 2 \sum_{l=0}^{\tau-1} L_v^2 \| \boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t} \|^2. \tag{41}
$$

Substituting (37), (38), (39), (40), and (41) into (36), and the learning rates satisfy $\eta_u \leq \frac{1}{4\tau(1+\chi)L_u}$ and $\eta_v \leq \frac{1}{2\tau(1+\chi)L_v}$, we have

$$
F_k(\boldsymbol{u}_{k,t+1}, \boldsymbol{v}_{k,t+1}) - F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \leq \left(2(1+\chi)L_u \eta_u^2 \tau^2 - \frac{1}{2}\eta_u \tau \right) \| \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2
$$

$$
+ \left((1+\chi)L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau \right) \| \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2 + (3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v) \sum_{l=0}^{\tau-1} \| \boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t} \|^2
$$

$$
+ (3\eta_u \chi^2 L_u L_v + 2\eta_v L_v^2) \sum_{l=0}^{\tau-1} \| \boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t} \|^2 + \frac{5}{4}\eta_u \lambda^2 \sum_{l=0}^{\tau-1} \| \nabla L_k(\boldsymbol{u}_{k,t,l}) \|^2. \tag{42}
$$

Below we focus on bounding two terms in (42), i.e., $\sum_{l=0}^{\tau-1} \| \boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t} \|^2$ and $\sum_{l=0}^{\tau-1} \| \boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t} \|^2$. Firstly, for $\sum_{l=0}^{\tau-1} \| \boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t} \|^2$, we have

$$
\sum_{l=0}^{\tau-1} \| \boldsymbol{v}_{k,t,l} - \boldsymbol{v}_{k,t} \|^2 = \sum_{l=0}^{\tau-1} \eta_v^2 \left\| \sum_{n=0}^{l-1} \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) \right\|^2
$$

$$
\overset{(a)}{\leq} \sum_{l=0}^{\tau-1} \eta_v^2 l \sum_{n=0}^{l-1} \| \nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) \|^2 \overset{(b)}{\leq} \eta_v^2 G_2^2 \frac{\tau(\tau+1)(2\tau+1)}{6}, \tag{43}
$$

where (a) comes from the Jensen's inequality, (b) follows the bounded gradient assumption in Assumption 2. For $\sum_{l=0}^{\tau-1} \| \boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t} \|^2$, we have

$$
\sum_{l=0}^{\tau-1} \| \boldsymbol{u}_{k,t,l} - \boldsymbol{u}_{k,t} \|^2 = \sum_{l=0}^{\tau-1} \eta_u^2 \left\| \sum_{n=0}^{l-1} (\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,n})) \right\|^2
$$

$$
\overset{(a)}{\leq} \sum_{l=0}^{\tau-1} \eta_u^2 l \sum_{n=0}^{l-1} \| \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) + \lambda \nabla L_k(\boldsymbol{u}_{k,t,n}) \|^2
$$

$$
\overset{(b)}{\leq} \sum_{l=0}^{\tau-1} \eta_u^2 l \sum_{n=0}^{l-1} 2 \| \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t,n}, \boldsymbol{v}_{k,t,n}) \|^2 + \sum_{l=0}^{\tau-1} \eta_u^2 l \sum_{n=0}^{l-1} 2 \| \lambda \nabla L_k(\boldsymbol{u}_{k,t,n}) \|^2
$$

$$
\overset{(c)}{\leq} \frac{\tau(\tau+1)(2\tau+1)}{3} \eta_u^2 G_1^2 + 2\eta_u^2 \lambda^2 \sum_{l=0}^{\tau-1} l \sum_{n=0}^{l-1} \| \nabla L_k(\boldsymbol{u}_{k,t,n}) \|^2, \tag{44}
$$

where (a) is due to the Jensen's inequality, (b) follows the triangle inequlity, (c) is due to Assumption 2. Substituting (43) and (44) into (42), the proof is completed.

C. Proof of Theorem 1

By substituting (18) into (2), we have the one-round convergence bounded of the global loss as follows:

$$
F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}_t) \leq \sum_{k=1}^{K} \frac{D_k}{D} (2(1+\chi)L_u \eta_u^2 \tau^2 - \frac{1}{2}\eta_u \tau) \| \nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2
$$

$$
+ \sum_{k=1}^{K} \frac{D_k}{D} ((1+\chi)L_v \eta_v^2 \tau^2 - \frac{1}{2}\eta_v \tau) \| \nabla_v F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) \|^2 + \frac{5}{4}\eta_u \lambda^2 \sum_{k=1}^{K} \frac{D_k}{D} \sum_{l=0}^{\tau-1} \| \nabla L_k(\boldsymbol{u}_{k,t,l}) \|^2
$$

$$
+ A_1 + 2\eta_u^2 \lambda^2 (3\eta_u L_u^2 + 2\eta_v \chi^2 L_u L_v) \sum_{k=1}^{K} \frac{D_k}{D} \sum_{l=0}^{\tau-1} (\tau - l) \| \nabla L_k(\boldsymbol{u}_{k,t,l}) \|^2, \tag{45}
$$

Below we bound $\|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2$. For ease of proof, we introduce an auxiliary variable $\bar{\boldsymbol{\Omega}}_{c,t} = \frac{\sum_{k\in\mathcal{K}} D_{k,c}\boldsymbol{\Omega}_{k,c,t}}{\sum_{k\in\mathcal{K}} D_{k,c}}$, which aggregates all devices's knowledge about class c ($\forall c \in \mathcal{C}$).

$$
\begin{aligned}
\|\nabla L_k(\boldsymbol{u}_{k,t,l})\|^2 &= \left\| \frac{1}{D_k} \sum_{c=1}^{C} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \boldsymbol{\Omega}_{c,t}\| \nabla h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) \right\|^2 \\
&\overset{(a)}{\leq} \frac{1}{D_k^2} C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l};x) - \boldsymbol{\Omega}_{c,t}\|^2 \|\nabla h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x})\|^2 \\
&\overset{(b)}{\leq} \frac{1}{D_k^2} C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \boldsymbol{\Omega}_{c,t}\|^2 \vartheta^2 \\
&\overset{(c)}{\leq} 2\frac{1}{D_k^2} \vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \bar{\boldsymbol{\Omega}}_{c,t}\|^2 + 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c}^2 \|\bar{\boldsymbol{\Omega}}_{c,t} - \boldsymbol{\Omega}_{c,t}\|^2, \quad (46)
\end{aligned}
$$

where (a) follows Jensen's inequality, (b) is due to Assumption 3, (c) derived by adding and substracting $\bar{\boldsymbol{\Omega}}_{c,t}$ into $\boldsymbol{\Omega}_{c,t}$ and using the triangle inequality.

Below we focus on bounding the two terms on the RHS of (46), where the first term is bounded as

$$
\begin{aligned}
&2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - \bar{\boldsymbol{\Omega}}_{c,t}\|^2 \\
&= 2\frac{1}{D_k^2}\vartheta^2 C \sum_{c=1}^{C} D_{k,c} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_{k,c}} \left\| \frac{1}{D_c} \sum_{h=1}^{K} \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{n,c}} (h_k(\boldsymbol{u}_{k,t,l};\boldsymbol{x}) - h_n(\boldsymbol{u}_{h,t};\boldsymbol{x}_1)) \right\|^2 \\
&\leq 8\vartheta^2\varsigma^2, \quad (47)
\end{aligned}
$$

where the inequality is due to Jensen's inequality and Assumption 3. For the second term on the RHS of (46), we have

$$
\begin{aligned}
\|\bar{\boldsymbol{\Omega}}_{c,t} - \boldsymbol{\Omega}_{c,t}\|^2 &= \left\| \frac{\sum_{k=1}^{K} \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c} - \frac{\sum_{k=1}^{K} \alpha_{k,t-1} \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{\sum_{k=1}^{K} \alpha_{k,t-1} D_{k,c}} \right\|^2 \\
&= \left\| \frac{\sum_{k=1}^{K}(1-\alpha_{k,t-1}) \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c} - \frac{(D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c})\sum_{k=1}^{K}\alpha_{k,t-1}\sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)}{D_c \sum_{k=1}^{K} \alpha_{k,t-1} D_{k,c}} \right\|^2 \\
&\leq \left(\frac{\sum_{k=1}^{K}(1-\alpha_{k,t-1}) \sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)\|}{D_c} + \frac{(D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c})\sum_{k=1}^{K}\alpha_{k,t-1}\sum_{(\boldsymbol{x}_1,y_1)\in\mathcal{D}_{k,c}} \|h_k(\boldsymbol{u}_{k,t};\boldsymbol{x}_1)\|}{D_c \sum_{k=1}^{K} \alpha_{k,t-1}\mathcal{D}_{k,c}} \right)^2 \\
&\overset{(a)}{\leq} 4\varsigma^2 \left(\frac{D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c}}{D_c} \right)^2, \quad (48)
\end{aligned}
$$

where (a) is due to Assumption 3. Substituting (46), (47), and (48) into (45), then substracting $F(\boldsymbol{W}^*)$ into both $F(\boldsymbol{W}_{t+1})$ and $F(\boldsymbol{W}_t)$, we have

$$
\begin{aligned}
F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}^*) &\leq F(\boldsymbol{W}_t) - F(\boldsymbol{W}^*) + \left(2(1+\chi)L_u\eta_u^2\tau^2 - \frac{1}{2}\eta_u\tau \right) \sum_{k=1}^{K} \frac{D_k}{D} \|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \\
&+ \left((1+\chi)L_v\eta_v^2\tau^2 - \frac{1}{2}\eta_v\tau \right) \sum_{k=1}^{K} \frac{D_k}{D} \|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t},\boldsymbol{v}_{k,t})\|^2 \\
&+ A_1 + A_2 + A_2 \sum_{k=1}^{K} \frac{1}{D}\frac{1}{D_k}C \sum_{c=1}^{C} D_{k,c}^2 \left(\frac{D_c - \sum_{k=1}^{K}\alpha_{k,t-1}D_{k,c}}{D_c} \right)^2, \quad (49)
\end{aligned}
$$

where $A_2 = 10\eta_u\lambda^2\tau\vartheta^2\varsigma^2 + 8\eta_u^2\lambda^2\vartheta^2\varsigma^2 (3\eta_u L_u^2 + 2\eta_v\chi^2 L_u L_v)\tau(\tau+1)$.

By using the L-smooth of loss functions, we have

$$\|\nabla_{\boldsymbol{u}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq 2L_u\left(F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) - F_k(\boldsymbol{u}_k^*, \boldsymbol{v}_k^*)\right), \tag{50}$$

and

$$\|\nabla_{\boldsymbol{v}} F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t})\|^2 \leq 2L_v\left(F_k(\boldsymbol{u}_{k,t}, \boldsymbol{v}_{k,t}) - F_k(\boldsymbol{u}_k^*, \boldsymbol{v}_k^*)\right). \tag{51}$$

Substituting (50) and (51) into (49), we have

$$F(\boldsymbol{W}_{t+1}) - F(\boldsymbol{W}^*) \leq A_3(F(\boldsymbol{W}_t) - F(\boldsymbol{W}^*)) + A_1 + A_2$$
$$+ A_2 \sum_{k=1}^{K} \frac{1}{D}\frac{1}{D_k} C \sum_{c=1}^{C} D_{k,c}^2 \Big(\frac{D_c - \sum_{k=1}^{K} \alpha_{k,t-1}D_{k,c}}{D_c}\Big)^2, \tag{52}$$

where $A_3 = 1 + (4L_u^2\eta_u^2 + 2L_v^2\eta_v^2)(1+\chi)\tau^2 - (\eta_u L_u + \eta_v L_v)\tau$. By telescoping the above inequality, we have

$$F(\boldsymbol{W}_T) - F(\boldsymbol{W}^*) \leq A_3^T(F(\boldsymbol{W}_0) - F(\boldsymbol{W}^*)) + \frac{1-A_3^T}{1-A_3}(A_1 + A_2)$$
$$+ A_2 \sum_{t=1}^{T-1} A_3^{T-1-t} \sum_{k=1}^{K} \frac{1}{D}\frac{1}{D_k} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_c^2}\Big(D_c - \sum_{k=1}^{K} \alpha_{k,t-1}D_{k,c}\Big)^2. \tag{53}$$

Below we bounding the last term on the RHS of (53) as

$$A_2 \sum_{t=1}^{T-1} A_3^{T-1-t} \sum_{k=1}^{K} \frac{1}{D}\frac{1}{D_k} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_c^2}\Big(D_c - \sum_{k=1}^{K} \alpha_{k,t-1}D_{k,c}\Big)^2$$
$$= A_2 \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \frac{1}{D}\frac{1}{D_k} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_c^2}\Big(D_c - \sum_{k=1}^{K} \alpha_{k,t}D_{k,c}\Big)^2$$
$$\overset{(a)}{\leq} \frac{A_2 KC}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} (1-\alpha_{k,t})D_{k,c}^2$$
$$= \frac{A_2 CK}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} D_{k,c}^2 - \frac{A_2 K}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} \alpha_{k,t}D_{k,c}^2, \tag{54}$$

where (a) is due to Jensen's inequality and $(1-\alpha_{k,t})^2 = 1-\alpha_{k,t}$. For the last term on the RHS of (54), we have

$$\frac{A_2 K}{D} \sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} C \sum_{c=1}^{C} \frac{D_{k,c}^2}{D_k D_c^2} \sum_{k=1}^{K} \alpha_{k,t}D_{k,c}^2$$
$$\overset{(a)}{\geq} A_2 \frac{1}{DK(T-1)} \Big(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{c=1}^{C} \sum_{k=1}^{K} \frac{D_{k,c}}{\sqrt{D_k}D_c} \sum_{k=1}^{K} \alpha_{k,t}D_{k,c}\Big)^2$$
$$\geq A_2 \frac{1}{DK(T-1)} \frac{1}{\max_{1\leq k\leq K} D_k} \Big(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \alpha_{k,t} \sum_{c=1}^{C} D_{k,c}\Big)^2$$
$$= A_2 \frac{1}{DK(T-1)} \frac{1}{\max_{1\leq k\leq K} D_k} \Big(\sum_{t=0}^{T-2} A_3^{T-2-t} \sum_{k=1}^{K} \alpha_{k,t}D_k\Big)^2, \tag{55}$$

where (a) follows Jensen's inequality. Substituting (54) and (55) into (53), the proof is completed.

D. Proof of Proposition 2

For the ease of presentation, we define the Lyapunov function as $\mathcal{V}(t) = \sum_{k=1}^{K} \frac{1}{2}q_{k,t}^2$, the Lyapunov drift of round t as $\Delta_1(t) = \mathcal{V}(t+1) - \mathcal{V}(t)$. According to the evolution of the virtual queue defined in (21), we have $q_{k,t+1}^2 \leq \left(q_{k,t} + \alpha_{k,t}E_{k,t} - \frac{E_k}{T}\right)^2$. For $\Delta_1(t)$, we have

$$\Delta_1(t) = \frac{1}{2}\sum_{k=1}^{K}(q_{k,t+1}^2 - q_{k,t}^2) \leq \sum_{k=1}^{K}\Big(\frac{1}{2}(q_{k,t} + \alpha_{k,t}E_{k,t} - \frac{E_k}{T})^2 - \frac{1}{2}q_{k,t}^2\Big)$$
$$\leq \zeta_0 + \sum_{k=1}^{K} q_{k,t}\Big(\alpha_{k,t}E_{k,t} - \frac{E_k}{T}\Big), \tag{56}$$

where $\zeta_0 = \frac{1}{2}\sum_{k=1}^{K}\zeta_k^2$, $\zeta_k = \max_t\{|\alpha_{k,t}E_{k,t} - \frac{E_k}{T}|\}$. By adding $-V\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k$ on both sides of (56), an upper bound of the one-round drift-plus-penalty function is given by

$$\Delta_1(t) - V\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k \le \zeta_0 + \sum_{k=1}^{K}q_{k,t}(\alpha_{k,t}E_{k,t} - \frac{E_k}{T}) - V\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k. \quad (57)$$

The drift-plus-penalty algorithm of Lyapunov optimization aims to minimize the upper bound of $\Delta_1(t) - \gamma_t V\sum_{k=1}^{K}\alpha_{k,t}D_k$. Define the T-round drift as $\Delta_T = \mathcal{V}(T-1) - \mathcal{V}(0) = \sum_{k=1}^{K}\frac{1}{2}q_{k,T-1}^2$. Then, the T-round drift-plus-penalty function can be bounded by

$$\Delta_T - V\sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k \le T\zeta_0 + \sum_{t=0}^{T-1}\Big(\sum_{k=1}^{K}q_{k,t}(\alpha_{k,t}E_{k,t} - \frac{E_k}{T}) - V\gamma_t\sum_{k=1}^{K}\alpha_{k,t}D_k\Big). \quad (58)$$

Based on the above analysis, we first prove the feasibility of the proposed algorithm. We use superscript * to denote the optimal offline solution of problem $\widetilde{\mathcal{P}}$, superscript † to represent the solution of the proposed drift-plus-penalty algorithm. For a feasible solution with $\alpha_{k,t} = 0$ and $E_{k,t} = 0$, we have $\quad \Delta_T = \sum_{k=1}^{K}\frac{1}{2}q_{k,T-1}^2 \le T\zeta_0 + V\sum_{t=0}^{T-1}\gamma_t D. \quad (59)$

Thus, we have
$$\Big(\sum_{k=1}^{K}q_{k,T-1}\Big)^2 \le K\sum_{k=1}^{K}q_{k,T-1}^2 \le 2K\Big(T\zeta_0 + V\sum_{t=0}^{T-1}\gamma_t D\Big), \quad (60)$$

where the first inequation comes from Jensen's inequality. According to the evolution of the virtual queue defined in (21), we have $\alpha_{k,t}E_{k,t} - \frac{E_k}{T} \le q_{k,t+1} - q_{k,t}$, summing this inequation over T rounds, we have

$$\sum_{t=0}^{T-1}\sum_{k=1}^{K}\Big(\alpha_{k,t}E_{k,t} - \frac{E_k}{T}\Big) \le \sum_{t=0}^{T-1}\sum_{k=1}^{K}(q_{k,t+1} - q_{k,t}) \le \sqrt{2K\Big(T\zeta_0 + V\sum_{t=0}^{T-1}\gamma_t D\Big)}. \quad (61)$$

By rearranging the above inequation, the energy consumption bound in (31) is derived. Below we analyze the optimality of the proposed drift-plus-penalty algorithm, which minimize the RHS in (58). Since Δ_T is positive, based on (58), we have

$$-V\sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}^{\dagger}D_{k,c} \le T\zeta_0 + \sum_{t=0}^{T-1}\sum_{k=1}^{K}q_{k,t}\Big(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k^{*}}{T}\Big) - V\sum_{t=0}^{T-1}\gamma_t\sum_{k=1}^{K}\alpha_{k,t}^{*}D_{k,c}, \quad (62)$$

Next, we bound the second term in the RHS of (62) as

$$\sum_{t=0}^{T-1}\sum_{k=1}^{K}q_{k,t}\Big(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k}{T}\Big) = \sum_{t=0}^{T-1}\sum_{k=1}^{K}(q_{k,t} - q_{k,0})\Big(\alpha_{k,t}^{*}E_{k,t} - \frac{E_k}{T}\Big) \le \frac{T(T-1)}{2}\sum_{k=1}^{K}\zeta_k^2. \quad (63)$$

Substituting (63) into (62), the inequation (30) is derived, and the proof is completed.

REFERENCES

[1] Z. Chen, W. Yi, Y. Liu, and A. Nallanathan, "Communication-efficient federated learning with heterogeneous devices," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2023.
[2] X. Huang, K. Zhang, F. Wu, and S. Leng, "Collaborative machine learning for energy-efficient edge networks in 6G," *IEEE Netw.*, vol. 35, no. 6, pp. 12–19, 2021.
[3] 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.
[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, June 2018.
[5] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6g," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 140–145, 2022.
[6] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, 2021.
[7] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, 2021.

[8] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, and S. Cui, "Communication-efficient federated edge learning via optimal probabilistic device scheduling," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 8536–8551, 2022.

[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] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2150–2167, 2021.

[11] Y.-S. Jeon, M. M. Amiri, J. Li, and H. V. Poor, "A compressive sensing approach for federated learning over massive mimo communication systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1990–2004, 2021.

[12] 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.

[13] S. Chen, C. Shen, L. Zhang, and Y. Tang, "Dynamic aggregation for heterogeneous quantization in federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6804–6819, 2021.

[14] 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.

[15] 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.

[16] 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.

[17] Z. Chen, W. Yi, A. Nallanathan, and G. Y. Li, "Federated learning for energy-limited wireless networks: A partial model aggregation approach," *arXiv preprint arXiv:2204.09746*, 2022.

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv*, 2015.

[19] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Communication-efficient federated distillation with active data sampling," in *Proc. IEEE Int. Conf. Commun.(ICC)*, 2022.

[20] 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.

[21] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learning (ICML)*, 18–24 Jul, 2021.

[22] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE Annual Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC)*, 2019, pp. 1–6.

[23] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2020.

[24] 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.

[25] T. Yu, E. Bagdasaryan, and V. Shmatikov, "Salvaging federated learning by local adaptation," *arXiv preprint arXiv:2002.04758*, 2020.

[26] I. Frades and R. Matthiesen, "Overview on techniques in cluster analysis," *Bioinformatics methods in clinical research*, pp. 81–107, 2010.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[28] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 20–22, Apr. 2017.

[29] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2019, pp. 691–706.

[30] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. ACM SIGSAC conf. comput. and commun. secur.*, 2017, pp. 603–618.

[31] E. Abbasnejad, J. Shi, and A. van den Hengel, "Deep Lipschitz networks and dudley GANs," 2018. [Online]. Available: https://openreview.net/forum?id=rkw-jlb0W

[32] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Proc. Neural Information Processing Systems (NeurIPS)*, 2018.

[33] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2020.

[34] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[35] Z. Chen, W. Yi, and A. Nallanathan, "The proofs and additional experimental results in the paper titled "resource-constrained heterogeneous wireless federated learning: A knowledge aggregation perspective"," *arXiv preprint, arXiv:2209.12277*, 2022. [Online]. Available: https://arxiv.org/abs/2209.12277

[36] 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.*, pp. 1–1, 2022.

[37] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[38] Z. Chen, Z. Zhou, and C. Chen, "Code caching-assisted computation offloading and resource allocation for multi-user mobile edge computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4517–4530, 2021.

[39] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Proc. Int. Conf. Mach. Learning (ICML)*, 18–24 Jul 2021.

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

[41] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[42] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, 2020.