IEEE ComSoc IEEE Communications Society
IEEE Open Journal of the Communications Society

Received 9 February 2026; revised 1 March 2026; accepted 12 March 2026. Date of publication 16 March 2026; date of current version 1 April 2026.

Digital Object Identifier 10.1109/OJCOMS.2026.3674612

Energy-Efficient and Learning-Aware Edge Intelligence With Probabilistic Scheduling

HAIHUI XIE[1,2], PINGPING XU[1], ZHAOGANG SHU[1,2] (Senior Member, IEEE), SHUWU CHEN[1,2], AND CHANGSHENG YOU[3] (Member, IEEE)

[1]College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[2]Engineering Research Center of Smart and Agricultural Chip, Fujian Province University, Fuzhou 350002, China
[3]Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China

CORRESPONDING AUTHOR: S. CHEN (chenshuwu@fafu.edu.cn)

This work was supported in part by the Project of Industry-University-Institute Cooperation in Colleges and Universities in Fujian Province of China under Grant 2024H6007, in part by the Project of Industry-University-Institute Joint Innovation in Colleges and Universities in Fujian Province of China under Grant 2024H6030, in part by the Project of Fu-Xia-Quan Collaborative Innovation Platform in Fujian Province of China under Grant 2025E3012, and in part by the Overseas Research and Further Education Program for Young and Middle-Aged Backone Teachers from Universities in Fujian Province of China.

ABSTRACT In edge intelligence networks, various applications exist based on different users and datasets. To realize intelligence, many works design various edge learning models to process these data. However, different types of users and models have negative influence on the learning performance, even degrading the learning gain. To address these issues, we propose a learning-aware optimization model with probabilistic scheduling. First, we design users scheduling to reveal the relationships between resource allocation and learning performance. Then, we also develop a fast proximal linearized algorithm in the large-scale communication networks. Next, we derive tasks scheduling in asymptotic cases to improve resource utilization efficiency. At last, extensive experimental results corroborate that probabilistic scheduling and multi-objective learning models improve the distinct edge learning gain efficiently compared with the state-of-the-art benchmark algorithms.

INDEX TERMS Edge intelligence, probabilistic scheduling, parallel frameworks, fast proximal linearized algorithms.

I. INTRODUCTION

WITH the rapid growth of the Internet-of-Things (IoT), massive machine-type communication network is a main connection of IoT devices, edge servers, and cloud platforms, enabling collaboration over edge infrastructures. However, maintaining stable communication in large-scale IoT networks requires efficient resource management, such as low-latency communication, energy-efficient optimization, adaptive device scheduling, and sophisticated load balancing mechanisms [1], [2], [3]. While various IoT devices generate massive and heterogeneous data, transmitting these amounts of raw data not only results in a heavy communication burden but also decreases resource utilization efficiency [4], [5]. This has given rise to edge intelligence, integrating learning models with edge computing techniques [6].

However, edge intelligence platforms face significant challenges in data collection under resource-limited networks. Efficient data collection requires intelligent processing to obtain quality while minimizing resource consumption. In the edge networks, processing various data is a significant challenge in edge learning systems [7], [8]. While they are easy to implement, such systems are usually slower in adapting to various data types. Therefore, multi-objective learning tasks are used commonly to involve optimizing various conflicting objectives and balance different learning tasks. This requires the need that balances learning accuracy, resource utilization efficiency and so on, adaptive for edge networks.

A. RELATED WORKS AND MOTIVATION

In the literature, there have been some prior works investigating the communication-constrained schemes from different perspectives. In conventional wireless communication systems, the sum-rate maximization and max-min fairness schemes are two typical schemes, allocating resource efficiently under the throughput maximization principles [9], [10], [11]. Nonetheless, these two schemes only consider the wireless channel state information and do not account for the machine learning factors, so they may result in the poor learning performance. To address this issue, [12], [13] proposed a learning centric power allocation (LCPA) model. The LCPA model involves multi-objective learning tasks to process various data and provide a trade-off between communication and

learning. Moreover, [14] also designed a task-oriented computation scheme to maximize the gain between sensing, and communication, and computation. Nevertheless, these works lack an analysis of scenarios where learning is integrated with dense IoT networks.

In a dense IoT network, co-channel interference (CCI) is an important factor affecting resource allocation, since activating all connectivity at the same time would yield poor data rates due to the interference among large-scale networks' communications. Thus, the works [13], [15], [16], [17] have proposed scheduling metrics to decrease the CCI influence. However, these scheduling algorithms lack deep integration between data learning and the dense IoT network, degrading communication and learning efficiency.

To address these issues, we propose a resource allocation model with probabilistic scheduling. This optimization model mainly includes a learning model fitting process, probabilistic scheduling strategy, and power allocation. Specifically, since the multi-objective optimization model involves various learning tasks, the proposed optimization objective is fitted by historical datasets to guide the IoT-edge communications under learning-oriented principles. A probabilistic scheduling strategy is adopted to reinforce the coupling between leaning and resource allocation. On the one hand, we derive an users scheduling for a single-objective optimization to improve learning performance, and also design a fast proximal linearized algorithm with probabilistic scheduling (FPLA-PSH) for solving large-scale optimization efficiently. On the other hand, we design a multi-objective optimization model with tasks scheduling in asymptotic orthogonality cases to improve resource utilization efficiency. However, this multi-objective optimization model is a non-convex and non-smooth problem due to the shared nature of wireless power constraints, it is hard to jointly optimize various learning tasks and resource allocation. Therefore, it is essential to require a fundamental decomposition for parallel optimization.

B. SUMMARY OF RESULTS

In this paper, we mainly consider multi-objective learning-oriented problems with probabilistic scheduling. By utilizing the alternating direction method of multipliers (ADMM), we first develop a fast proximal linearized algorithm with probabilistic scheduling. Next, we design a parallel algorithm to solve the multi-objective learning case. Lastly, extensive experimental results demonstrate the performance of the proposed schemes. In brief, the significant contributions of the paper are summarized as follows:

- A learning-oriented optimization model with probabilistic scheduling is proposed to guide communication-efficient data collection. The optimization objective is fitted from large amount of historical datasets to predict the optimized resources for learning tasks.
- A FPLA algorithm with users scheduling is developed to establish the relationship between learning and resource allocation. Moreover, a fast proximal linearized algorithm is designed to solve large-scale problems.

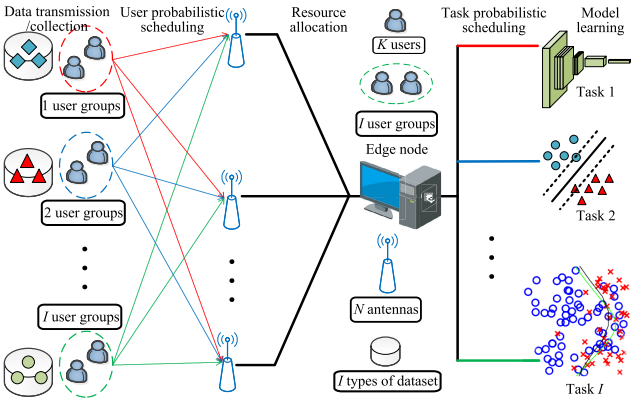


FIGURE 1. System model of probabilistic scheduling at the edge.

- A parallelized algorithm with tasks scheduling is derived to improve resource utilization efficiency. Specifically, decomposing and eliminating auxiliary variables realizes highly parallelization and ensure convergence.

C. OUTLINE

The remainder of this paper is organized as follows. Section II describes the system model and formulates a multi-objective optimization problem. Section III and Section IV design FPLA and parallel algorithms to solve large-scale and multi-objective optimization subproblems, respectively. Section V discusses the experimental results, and finally, Section VI draws the conclusion.

Notations: Scalars are denoted by italic regular letters while vectors and matrices are denoted by lower- and upper-case bold letters, respectively. $\boldsymbol{D}$ is a diagonal matrix whose diagonal entries are specified by the diagonal ones of channel gain matrix $\boldsymbol{G}$. The symbol $\mathbf{1}$ indicates a vector with all entries being unity, and $\mathbf{0}$ represents a vector with all entries being zeros. $\mathcal{CN}(\mathbf{0}, \varrho_k \boldsymbol{I})$ stands for a complex Gaussian distribution with mean vector $\mathbf{0}$ and variance matrix $\varrho_k \boldsymbol{I}$. The distribution $\mathcal{B}(p)$ follows a Bernoulli distribution with probability p.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first design a data transmission pipeline at an edge node. Then, we analyze the probabilistic scheduling. Finally, we formulate a learning-oriented problem.

A. SYSTEM MODEL

As depicted in Fig. 1, the edge system consists of an edge node and a pool of potential users. Specifically, there exist a global task group $\mathcal{I}$ with I tasks, N antennas and a global user group $\mathcal{K}$ with K users. By I learning tasks, K users can be separated into I user groups $\{\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_I\}$, which can obtain I groups of various datasets at the same time. Overall, the pipeline encompasses data transmission, users probabilistic scheduling, resource allocation, tasks probabilistic scheduling, and various model learning. Different user groups generate various data and transmit these data to an edge node via the uplink of a massive connection, with each

group contributing to specific data. The edge server utilizes the collected data to train different models, containing convolutional neural network (CNN) [18], support vector machine (SVM) [19], 110-layer deep residual network (ResNet110) [20], and a cloud-point neural network (PointNet) [21].

There exist a massive connectivity optimization problem at this edge, so we mainly focus on the wireless connectivity scheduling policy of how efficiently activates this dense network for the optimized learning performance. Our goal is to choose a subset of connectivity to activate in any given transmission period so as to maximize some network utility function of the achieved long-term average rates [15]. In this respect, we first define $w_k \in \{0,\ 1\}$ as a scheduling indicator, where $w_k = 1$ represents the activated communication for user k and $w_k = 0$ otherwise. Therefore, the achievable rate R_k for the k^{th} user is given by

$$R_k = \log_2\left(1 + \frac{G_{k,k}p_k}{\sum\limits_{\ell\in\mathcal{K}\setminus k} w_k G_{k,\ell}p_\ell + \sigma^2}\right),\ \forall k \in \mathcal{K}_i. \quad (1)$$

where p_k denotes the transmit power of the k^{th} user, σ^2 denotes the variance of Gaussian noises, and $G_{k,\ell}$ represents the composite channel gain from the ℓ^{th} user to the edge server when decoding data of the k^{th} user, computed as $G_{k,k} = \rho_k\|\boldsymbol{h}_k\|_2^2$ if $\ell = k$, and $G_{k,\ell} = \rho_\ell|\boldsymbol{h}_k^H\boldsymbol{h}_\ell|^2/\|\boldsymbol{h}_k\|_2^2$ if $\ell \neq k$, with $\boldsymbol{h}_k \in \mathbb{C}^{N\times 1}$ being the complex-valued channel fast-fading vector from the k^{th} user to the edge server and ρ_k being the path loss of the k^{th} user. And then, the total number of samples obtained for the i^{th} task is

$$D_i^r = \sum_{k\in\mathcal{K}_i}\left\lfloor\frac{BTR_k}{V_i}\right\rfloor + A_i \approx \sum_{k\in\mathcal{K}_i}\frac{BTR_k}{V_i} + A_i, \quad (2)$$

where B is the bandwidth in Hz that is assigned to this system, T is the total transmission time in second, V_i is the number of bits for each data sample and A_i is the initial number of historical samples for task i.

Remark 1 (Parameter Fitting and Dada Shifts [12], [13]): Consider a sequence of training datasets $\{D_i^1,\ D_i^2, \cdots,\ D_i^Q\}$. For each dataset D_i^m, we train the model to convergence and record the resulting loss ℓ_m. Using the pairs $\{D_i^m,\ \ell_m\}_{m=1}^Q$, the parameters $(a_i,\ b_i)$ of the expected learning loss model, which is typically chosen based on empirical observations in similar settings,

$$\Theta_i(D_i^m \mid a_i, b_i) \approx a_i\left(D_i^m\right)^{-b_i} \quad (3)$$

are fitted by minimizing the nonlinear least squares error:

$$\min_{a_i>0, b_i>0}\frac{1}{Q}\sum_{m=1}^{Q}\left|\ell_m - \Theta_i\left(D_i^m \mid a_i, b_i\right)\right|^2. \quad (4)$$

The learning error $\Theta_i(D_i^m \mid a_i, b_i)$ is universal for all deep learning models, satisfying the following properties: percentage, monotonic decreasing, and gradient decreasing. From the statistical perspective, a can be given by

$$a_i = \left(\frac{T_g}{2} + \frac{\mathrm{Tr}(\boldsymbol{U}_i\boldsymbol{V}_i^{-1})}{2W_i}\right)W_i,$$

where W_i is the number of parameters, T_g is the temperature parameter from the Gibbs distribution formulation, and $\boldsymbol{U}_i$, $\boldsymbol{V}_i$ contain the second-order and first-order derivatives of the generalization error. As the number of training samples goes to infinity, the weighting parameter a_i depends on the model complexity. While $b_i \to 1$ holds as $D_i^m \to \infty$, b_i can be tuned to capture dataset-specific correlations in finite-sample regimes.

While sudden or gradual distribution shifts at edge nodes happen, the dataset $\mathcal{D}$ is transformed as a new dataset $\mathcal{D}'\}$, as the same likelihood function $\ln p(\mathcal{D}'|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ denote prior, mean, and covariance distributions, respectively, and we can maximize the likelihood function $\ln p(\mathcal{D}'|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ to obtain the optimization parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, being adaptive to sudden or gradual distribution shifts. Sampling a sequence of new training samples $\{(D_i^m)', \ell_m'\}_{m=1}^Q$ from the new dataset $\mathcal{D}'$, the parameters $(a_i',\ b_i')$ of the expected learning loss model can be tuned by minimizing the nonlinear least squares error (4).

B. PROBLEM FORMULATION

This paper aims to investigate the multi-objective learning tasks and large-scale communication networks' influence on learning performance, by jointly optimizing efficient resource allocation and multi-objective learning tasks. We first employ (3) as an objective to minimize the learning errors, where the tuning parameters (a_i, b_i) consider the model complexity and the non-independent-and-identically dataset, respectively. Also, we use (3) to capture the learning loss shape [22], [23]. Specifically, we obtain (a_i, b_i) via fitting the learning error to match the training data very well [12]. Given the learning parameters $\{a_i,\ b_i\}$, we then formulate a resource allocation problem:

$$\mathcal{P}1: \min_{\boldsymbol{p},\ \{\mathcal{U}_{k,i}\}_{k\in\mathcal{K},\ i\in\mathcal{I}}}\ \max_{\omega_i\in\{0,\ 1\},\ i\in\mathcal{I}}\ \rho_i \times a_i\left(D_i^r\right)^{-b_i}$$

$$\text{s.t.}\ \sum_{k\in\mathcal{K}} w_k p_k = P,\ p_k \geq 0,\ k \in \mathcal{K}, \quad (5a)$$

$$w_k \in \{0,\ 1\},\ k \in \mathcal{K}, \quad (5b)$$

$$\omega_i \in \{0,\ 1\},\ i \in \mathcal{I}, \quad (5c)$$

$$\mathcal{U}_{k,\,i} = \mathbb{P}(R_k > \gamma,\ w_k = 1,\ \omega_i = 1), \quad (5d)$$

where ρ_i is a weighting factor ($\rho_i \geq 1$) which approximates the learning error. (5a) is obtained by [12], since a larger $\sum_{k\in\mathcal{K}} w_k p_k$ always improves the performance. As shown in (5b) and (5c), the scheduling indicators $w_k = 1$ and $\omega_i = 0$ represents that user k is chosen by learning task i for transmission and $w_k = 0$ or $\omega_i = 0$ otherwise. Moreover, (5d) is the scheduling probability to deliver sufficient transmission performance, where w_k denotes a random variable

following a Bernoulli distribution $\mathcal{B}(\mathcal{U}_{k,i})$. To successfully decode the updates from user k, the achievable rate $R_k > \gamma$ and $w_k = 1$ represent the influence of channel gain and participating users on learning performance, respectively.

Remark 2 (Probabilistic Scheduling Policy): Traditional scheduling algorithms only depend on channel state information and fail to account for the impact of node location distribution on learning performance [15], [16]. To deal with this issue, we design a probabilistic scheduling approach, incorporating channel distribution variations and leveraging the spatial distribution of users' locations. During a communication round t, to update the power parameter p_k from user k to the edge, user k should be scheduled by the edge. The probability of a generic scheduled depends on a number of stochastic processes, e.g., the random spatial distribution of the edge/user locations and small-scale fading, thus we define (5d), termed the scheduling probability, to characterize the transmission performance.

Compared with [13], $\mathcal{P}1$ involves various learning tasks with probabilistic scheduling policy, which includes users scheduling and tasks scheduling. To get insight on the relationship between scheduling and learning, we analyze single-task and asymptotic two specific cases to design efficient resource allocation algorithms. On the one hand, we consider a single task case, i.e., $\omega_1 = 1$ holds, (5d) is rewritten as

$$\mathcal{U}_k = \mathbb{P}(R_k > \gamma,\ w_k = 1), \tag{6}$$

where $\mathcal{U}_k$ denotes the successful transmission of each user. On the other hand, we consider the asymptotic case, i.e., $w_k = 1$ and $R_k > \gamma$ hold, then (5d) is rewritten as

$$\mathcal{U}_i = \mathbb{P}(\omega_i = 1). \tag{7}$$

The scheduling probability of task i is also given by

$$\lambda_i = \sum_{i=1}^{I} \omega_i \mathcal{U}_i. \tag{8}$$

Therefore, we next derive users and tasks scheduling cases to design resource allocation algorithms, respectively.

III. USERS SCHEDULING CASE: FAST PROXIMAL LINEARIZED ALGORITHM

In this section, we first design an users scheduling. Then, we propose a fast proximal linearized algorithm.

A. USERS SCHEDULING

We have formulated a multi-objective optimization problem $\mathcal{P}1$. However, $\mathcal{P}1$ exhibits strong coupling between the learning and communication objectives. Moreover, its use of auxiliary variables and linearized constraint approximations tends to slow down convergence in large-scale networks. To address these issues, we reformulate the problem into a single-objective learning-oriented form and propose a variable decomposition technique for efficient solving. Based on $\mathcal{P}1$, we first formulate this single-objective optimization problem as

$$\mathcal{P}2: \min_{\boldsymbol{p},\,\{\mathcal{U}_k\}_{k\in\mathcal{K}}} \ \rho \times a\left(D^r\right)^{-b}$$
$$\text{s.t. (5a), (5b), (5d).}$$

Since $\mathcal{P}2$ is still non-convex, we introduce additional variables $\boldsymbol{\delta}$ to separate interference terms and then rewrite $\mathcal{P}2$ as

$$\mathcal{P}3: \min_{\boldsymbol{p},\,\boldsymbol{\delta},\,\{\mathcal{U}_k\}_{k\in\mathcal{K}}} \ \Phi(\boldsymbol{p}|\boldsymbol{\delta})$$
$$\text{s.t.} \ \sum_{\ell\in\mathcal{K}\setminus k} w_\ell G_{k,\ell} p_\ell + \sigma^2 = \delta_k,$$
$$\text{(5a), (5b), (5d),} \tag{9a}$$

with

$$\Phi(\boldsymbol{p}|\boldsymbol{\delta}) \triangleq \rho \times a\left(\sum_{k\in\mathcal{K}} \frac{BT}{V} w_k \log_2\left(1+\frac{G_{k,k}p_k}{\delta_k}\right)+A\right)^{-b}.$$

However, it is highly challenging for $\mathcal{P}3$ to optimize a distribution. Thus, by deriving the relationship between the distribution and the learning objective, and by leveraging its effect on the learning rate, we obtain a distributed solution to $\mathcal{P}3$. We assume that the updates are independent-and-identically distributed and using the law of large numbers, and then obtain the expression of (6) as follows:

$$\mathcal{U}_k = \lim_{t\to\infty} \underbrace{\frac{1}{Kt}\sum_{\ell=0}^{t-1}\sum_{k=1}^{K} \mathbb{1}\left\{w_k(\ell)=1,\ R_k(\ell)>\gamma\right\}}_{\nu(t)}, \tag{10}$$

where $\nu(t)$ is defined as the updating step for scheduling variables $\boldsymbol{p}$. To establish the relationship between the scheduling probability and learning error in $\mathcal{P}3$, we derive Theorem 1 as follows.

Theorem 1 (Bounded learning errors): For any convergence precision ϵ, the learning error under the probabilistic scheduling policy enables to achieve an ϵ gap after T rounds of communication, i.e.,

$$\Phi(\boldsymbol{p}(T)|\boldsymbol{\delta}(T)) - \Phi(\boldsymbol{p}^*, \boldsymbol{\delta}^*) \le \epsilon, \tag{11}$$

if T is satisfied as follows:

$$T \ge \frac{\log(\epsilon/I)}{\log(1-(1-\epsilon_n)\mathcal{U}_k)}, \tag{12}$$

where $\epsilon_n = \min(\Phi(\boldsymbol{p}(T)|\boldsymbol{\delta}(T)), 1) \in (0,1)$ denotes the error level of $\mathcal{P}1$, and $\boldsymbol{p}^*, \boldsymbol{\delta}^*$ represent the optimal variables when $\mathcal{P}1$ is minimized.

Proof: See Appendix A. ■

Theorem 1 demonstrates the general convergence property of learning in wireless networks. Given convergence precision ϵ, optimizing this probability distribution (10) is able to reduce the number of learning iterations required. However, (10) represents a limiting expression and general formula that are computationally intractable. Therefore, combining (10)

with the definition of $\nu(t)$, we make the following approximation of $\nu(t)$ in practice.

$$\nu(t+1) = \frac{t}{t+1}\nu(t) + \frac{\sum_{k=1}^{K} \mathbb{1}\{w_k(t)=1,\ R_k(t)>\gamma\}}{K(t+1)}. \tag{13}$$

We next develop an unbiased estimation framework based on Assumption 1 to analyze the validity of (13).

Assumption 1 (Stability assumption [17]): The joint process $\{\mathbb{1}\{w_k(t)=1, R_k(t)>\gamma\}\}_{k=1}^{K}$ is stationary.

Assumption 1 is standard in the analysis of stochastic network optimization and holds under common fading channel models with stationary scheduling policies. Given Assumption 1, we give the unbiased estimation result of (13) as follows.

Proposition 1 (Unbiased estimation): Under Assumption 1 and the recursive equation (13), $\mathcal{U}_k$ defined in (10) is equal to the steady-state expectation of $\nu(t)$ and the instantaneous update ratio, i.e.,

$$\mathcal{U}_k = \mathbb{E}[\nu(t)] = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\mathbb{1}\{w_k(t)=1, R_k(t)>\gamma\}\right]. \tag{14}$$

Proof: See Appendix C. ∎

From Proposition 1, it can be concluded that (13) is an unbiased estimator for (10). It allows us to shift our analytical focus from the limiting time average in (10), which is defined over an infinite horizon, to the steady-state expectation in (14). This expectation is often more tractable for subsequent optimization and performance analysis using tools from stochastic optimization and queuing theory.

Remark 3 (The updating step): The equation (13) defines the factor $\nu(t)$ as a time-averaged estimator of the parameter update success probability, which is iteratively refined using the status of each transmission. Under favorable channel conditions, user updates are consistently received, leading to an increase in $\nu(t)$. Conversely, in disadvantageous communication scenarios, the value of $\nu(t)$ adaptively declines. This reduction deliberately imposes a more conservative update process. The mechanism is that unreliable communications yield fewer successful updates to the edge server, consequently causing only minor adjustments to the global variable. Governed by this lowered $\nu(t)$, local updates are restrained from abrupt changes, thereby maintaining alignment with the global state.

To sum up, the probabilistic scheduling decision is analytically determined, with extremely low computational complexity $\mathcal{O}(1)$. Moreover, the binary variable $\boldsymbol{w}$ is generated by Bernoulli distribution $\mathcal{B}(\nu(t+1))$ in the practical applications. Also, it is noted that our probabilistic scheduling strategy is fair with respect to different users.

B. FAST PROXIMAL LINEARIZED ALGORITHM

We have derived the relationship between the probability distribution and the learning rate, and decoupled the influence of scheduling. However, when the number of users scales to a large size, conventional ADMM algorithm severely increases computational complexity. To address this issue, we design a fast proximal linearized algorithm, to efficiently solve $\mathcal{P}3$ [24]. For a detailed exposition of the FPLA algorithm, we first present Assumption 2.

Assumption 2 (Smoothness[1] [12], [13]): The function $\Phi(\boldsymbol{p}|\boldsymbol{\delta})$ satisfies the following conditions:

1) $\Phi(\boldsymbol{p}|\boldsymbol{\delta}^*)$ is L_p-smooth, i.e., $\|\nabla_{\boldsymbol{x}}\Phi(\boldsymbol{x}|\boldsymbol{\delta}^*)-\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{y}|\boldsymbol{\delta}^*)\|_2 \le L_p\|\boldsymbol{x}-\boldsymbol{y}\|_2$ for any $\boldsymbol{x},\boldsymbol{y}$. In particular, we set $L_p = \lambda_{\max}\left(\nabla_{\boldsymbol{x}}^2\Phi(\boldsymbol{x}|\boldsymbol{\delta}^*)\right)$, where $\lambda_{\max}(\boldsymbol{X})$ represents the largest eigenvalue of $\boldsymbol{X}$.
2) $\Phi(\boldsymbol{p}^*|\boldsymbol{\delta})$ is L_δ-smooth, i.e., $\|\nabla_{\boldsymbol{x}}\Phi(\boldsymbol{p}^*|\boldsymbol{x})-\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{p}^*|\boldsymbol{y})\|_2 \le L_\delta\|\boldsymbol{x}-\boldsymbol{y}\|_2$ for any $\boldsymbol{x},\boldsymbol{y}$. In particular, we set $L_\delta = \lambda_{\max}\left(\nabla_{\boldsymbol{x}}^2\Phi(\boldsymbol{p}^*|\boldsymbol{x})\right)$.

where $\boldsymbol{\delta}^*$ and $\boldsymbol{p}^*$ denote the current values stored, respectively. Assumption 2 establishes the Lipschitz smoothness of $\Phi(\boldsymbol{p}|\boldsymbol{\delta})$ with respect to both optimization variables, which enables the application of gradient-based acceleration techniques to its minimization. Based on this property, we next derive the Augmented Lagrangian Function (ALF) for Problem $\mathcal{P}2$ in Proposition 2.

Proposition 2 (The ALF of $\mathcal{P}2$): We can obtain two relaxing ALFs about $\boldsymbol{p}$ and $\boldsymbol{\delta}$, respectively.

$$\begin{aligned}
&L\left(\boldsymbol{p}|\nabla_{\boldsymbol{y}_p}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)), \boldsymbol{p}(t), \boldsymbol{z}_p(t), \boldsymbol{\delta}(t); \boldsymbol{\alpha}(t), \beta(t)\right)\\
&= \langle\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)), \boldsymbol{p}\rangle + \sum_{k\in\mathcal{K}}\alpha_k(t)\sum_{\ell\in\mathcal{K}\setminus k} w_\ell G_{k,\ell}p_\ell\\
&\quad + \mu(t)\left\langle\left(\sum_{k\in\mathcal{K}} z_{p_k}(t) - P\right)\mathbf{1}, \boldsymbol{p}\right\rangle + \beta(t)\sum_{k\in\mathcal{K}} p_k\\
&\quad + \mu(t)\Big\langle(\boldsymbol{G}-\boldsymbol{D})^T\left((\boldsymbol{G}-\boldsymbol{D})(\widehat{\boldsymbol{w}}\bullet\boldsymbol{z}_p(t)) - \boldsymbol{z}_\delta(t) + \sigma^2\mathbf{1}\right),\\
&\quad \widehat{\boldsymbol{w}}\bullet\boldsymbol{p}\Big\rangle + \frac{1}{2}\left(L_p\theta(t) + \mu(t)\lambda_p\right)\left\|\boldsymbol{p}-\boldsymbol{z}_p(t)\right\|_2^2
\end{aligned} \tag{15}$$

and

$$\begin{aligned}
&L\left(\boldsymbol{\delta}|\nabla_{\boldsymbol{y}_\delta}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)), \boldsymbol{p}(t+1), \boldsymbol{\delta}(t), \boldsymbol{z}_\delta(t); \boldsymbol{\alpha}(t), \beta(t)\right)\\
&= \langle\nabla_{\boldsymbol{\delta}}\Phi\left(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)\right), \boldsymbol{\delta}\rangle - \sum_{k\in\mathcal{K}}\alpha_k(t)\delta_k\\
&\quad + \mu(t)\sum_{k\in\mathcal{K}}\delta_k\left(\sum_{\ell\in\mathcal{K}\setminus k} w_\ell G_{k,\ell}z_{p_\ell}(t) + \sigma^2 - z_{\delta_k}(t)\right)\\
&\quad + \frac{1}{2}\left(L_\delta\theta(t) + \mu(t)\lambda_\delta\right)\|\boldsymbol{\delta}-\boldsymbol{z}_\delta(t)\|_2^2,
\end{aligned} \tag{16}$$

where

$$\boldsymbol{y}_p(t+1) = (1-\theta(t))\boldsymbol{p}(t) + \theta(t)\boldsymbol{z}_p(t), \tag{17a}$$

$$\boldsymbol{y}_\delta(t+1) = (1-\theta(t))\boldsymbol{\delta}(t) + \theta(t)\boldsymbol{z}_\delta(t), \tag{17b}$$

$$\theta(t+1) = \frac{1}{2}\left(-\theta^2(t) + \sqrt{\theta^4(t) + 4\theta^2(t)}\right), \tag{17c}$$

with $\lambda_p \ge 2K\left(\|\boldsymbol{G}-\boldsymbol{D}\|_2+1\right)^2$, $\lambda_\delta \ge 2$, and the sequence of penalty parameters $\{\mu(t)\}$ given by $\mu(t) = 1/\theta(t)$.

[1]The Lipschitz smoothness holds due to its universal learning error model $a_i D_i^{-b_i}$ in the high volatility and device heterogeneity cases.

TABLE 1. Algorithm parameters and settings [24].

Algorithm parameters	Description and setting
Lipschitz constants L_p, L_δ	$L_p = \lambda_{\max}\left(\nabla^2_{\boldsymbol{x}} \Phi(\boldsymbol{x}\|\boldsymbol{\delta}^*)\right)$ $L_\delta = \lambda_{\max}\left(\nabla^2_{\boldsymbol{x}} \Phi(\boldsymbol{p}^*\|\boldsymbol{x})\right)$
The step size $\theta(t)$	$\theta(t) \in (0,\ 1) \leq C$ (C is a constant) $\sum \theta(t) = \infty,\ \sum \theta(t)^2 < 0$
Penalty parameters $\mu(t)$, $\beta(t)$	$\mu(t) = 1/\theta(t) \to \infty,\ \beta(t) > 0$
λ_p, λ_δ	$\lambda_p = 2K(\|\boldsymbol{G} - \boldsymbol{D}\|_2 + 1)^2,\ \lambda_\delta = 2$
Primal variables $\boldsymbol{p}$, $\boldsymbol{\delta}$	$\max(\|\boldsymbol{p}\|,\ \|\boldsymbol{\delta}\|) \leq C$
Auxiliary variables $\boldsymbol{y}_p$, $\boldsymbol{y}_\delta$, $\boldsymbol{z}_p$, $\boldsymbol{z}_\delta$	$\max(\|\boldsymbol{y}_p\|,\ \|\boldsymbol{y}_\delta\|,\ \|\boldsymbol{z}_p\|,\ \|\boldsymbol{z}_\delta\|) \leq C$
Dual variables $\boldsymbol{\alpha}(t)$	$\|\boldsymbol{\alpha}(t)\| \leq C$
Gradient $\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\|\boldsymbol{\delta}^*)$, $\nabla_{\boldsymbol{y}} \Phi(\boldsymbol{p}^*\|\boldsymbol{y})$	$\max(\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\|\boldsymbol{\delta}^*)\|,\ \|\nabla_{\boldsymbol{y}} \Phi(\boldsymbol{p}^*\|\boldsymbol{y})\|) \leq C$
Scheduling probability $\nu(t)$	$\nu(t) \in (0,\ 1) \leq C$

Proof: See Appendix C. ∎

Proposition 2 demonstrates that (17a) and (17b) accelerate the update of $\Phi(\boldsymbol{p}|\boldsymbol{\delta})$ by employing a look-ahead gradient mechanism [12]. Moreover, the quadratic regularizers $\big(L_p\theta(t) + \mu(t)\lambda_p\big)/2\left\|\boldsymbol{p} - \boldsymbol{z}_p(t)\right\|_2^2$ and $\big(L_\delta\theta(t) + \mu(t)\lambda_\delta\big)/2\|\boldsymbol{\delta} - \boldsymbol{z}_\delta(t)\|_2^2$ are introduced to linearize the augmented Lagrangian terms. To ensure the convergence result, we have now listed all required technical assumptions and added Table 1 to explain the meaning of important algorithmic parameters. As long as the parameters are set in Table 1, the optimization algorithm is guaranteed to converge. However, the linearization structures in Proposition 2 slow down the overall convergence. Therefore, we design an accelerated algorithm to address this issue [24]. Specifically, we first compute the partial derivatives with respect to the relevant variables, and then apply a gradient descent algorithm.

1) UPDATE OF $\boldsymbol{p}$ WITH FIXED VARIABLES

$$L\Big(\boldsymbol{p} \mid \nabla_{\boldsymbol{y}_p} \Phi\big(\boldsymbol{y}_p(t+1) \mid \boldsymbol{\delta}(t)\big), \boldsymbol{p}(t), \boldsymbol{z}_p(t), \boldsymbol{\delta}(t); \boldsymbol{\alpha}(t), \beta(t)\Big)$$

admits a closed-form solution with respect to $\boldsymbol{p}$. The solution is given by

$$\boldsymbol{z}_p(t+1) = \max\Big(\tilde{\boldsymbol{z}}_p(t+1) - \nu(t)\big(\boldsymbol{1} - \boldsymbol{w}(t)\big),\ \boldsymbol{0}\Big), \tag{18}$$

where each entry of $\boldsymbol{w}(t) \triangleq [w_1(t), \ldots, w_K(t)]^T$ follows a Bernoulli distribution $\mathcal{B}(\nu(t))$, and the k^{th} element of $\tilde{\boldsymbol{z}}_p(t+1)$ is given by

$$\begin{aligned}
&\tilde{z}_{p_k}(t+1) \\
&= \Bigg\{\Bigg(\sum_{\ell\in\mathcal{K}} \frac{BT}{V} w_\ell \log_2\Big(1 + \frac{G_{\ell,\ell} y_{p_\ell}(t+1)}{\delta_\ell(t)}\Big) + A\Bigg)^{-b-1} \cdot \\
&\frac{b\rho a BTw_k G_{k,k}}{\ln(2)V\big(\delta_k(t) + G_{k,k} y_{p_k}(t+1)\big)} - \sum_{\ell\in\mathcal{K}\setminus k} \alpha_\ell(t) w_k G_{\ell,k} \\
&\quad - \mu(t) \sum_{\ell\in\mathcal{K}\setminus k} \Big(\Big(\sum_{m\in\mathcal{K}\setminus\ell} w_m G_{\ell,m} z_{p_m}(t) \\
&\quad + \sigma^2 - z_{\delta_\ell}(t)\Big) w_k G_{\ell,k}\Big) \\
&\quad - \beta(t) - \mu(t)\Big(\sum_{k\in\mathcal{K}} z_{p_k}(t) - P\Big) \\
&\quad + \big(L_p\theta(t) + \mu(t)\lambda_p\big) z_{p_k}(t)\Bigg\} \\
&\Big/ \big(L_p\theta(t) + \mu(t)\lambda_p\big).
\end{aligned} \tag{19}$$

Combining (19) with the look-ahead gradient technique [12], we further update $\boldsymbol{p}$ as

$$\boldsymbol{p}(t+1) = \big(1 - \theta(t)\big)\boldsymbol{p}(t) + \theta(t)\boldsymbol{z}_p(t+1). \tag{20}$$

2) UPDATE OF δ WITH FIXED VARIABLES

The ALF of $\mathcal{P}2$ also admits a closed-form solution with respect to $\boldsymbol{\delta}$. The solution is given by

$$\boldsymbol{z}_\delta(t+1) = \max\big(\tilde{\boldsymbol{z}}_\delta(t+1),\ \sigma^2\boldsymbol{1}\big), \tag{21}$$

where the k^{th} element of $\tilde{\boldsymbol{z}}_\delta(t+1)$ can be computed as

$$\begin{aligned}
&\tilde{z}_{\delta_k}(t+1) \\
&= \Bigg\{\frac{-b\rho a BTV w_k G_{k,k} p_k(t+1)}{\ln 2}\big(y_{\delta_k}(t+1) + G_{k,k} p_k(t+1)\big) \\
&\quad \times y_{\delta_k}(t+1) \\
&\quad \times \Bigg(\sum_{\ell\in\mathcal{K}} \frac{BTw_\ell}{V} \log_2\Big(1 + \frac{G_{\ell,\ell} p_\ell(t+1)}{y_{\delta_\ell}(t+1)}\Big) + A\Bigg)^{-b-1} \\
&\quad - \alpha_k(t) - \mu(t)\Big(\sum_{\ell\in\mathcal{K}\setminus k} w_\ell G_{k,\ell} z_{p_\ell}(t) + \sigma^2 - z_{\delta_k}(t)\Big) \\
&\quad + \big(L_\delta\theta(t) + \mu(t)\lambda_\delta\big) z_{\delta_k}(t)\Bigg\} \Big/ \big(L_\delta\theta(t) + \mu(t)\big).
\end{aligned} \tag{22}$$

Combining (19) with the look-ahead gradient algorithm [12], we update $\boldsymbol{\delta}$ as

$$\boldsymbol{\delta}(t+1) = \big(1 - \theta(t)\big)\boldsymbol{\delta}(t) + \theta(t)\boldsymbol{z}_\delta(t+1). \tag{23}$$

3) UPDATE RELATIVE DUAL VARIABLES

We can obtain an optimized solution directly with respect to dual variables:

$$\begin{aligned}
\alpha_k(t+1) &= \alpha_k(t) + \mu(t)\Big(\sum_{\ell\in\mathcal{K}\setminus k} w_\ell G_{k,\ell} z_{p_\ell}(t+1) \\
&\quad + \sigma^2 - z_{\delta_k}(t+1)\Big),
\end{aligned} \tag{24a}$$

$$\beta(t+1) = \beta(t) + \mu(t)\Big(\sum_{k\in\mathcal{K}} z_{p_k}(t+1) - P\Big). \tag{24b}$$

It is obvious that optimizing power consumption and mitigating interference in large-scale networks improve the capability and quality of data collection at edge systems. In terms of computational complexity, it is evident that the FPLA algorithm exhibits a low per-iteration cost of $\mathcal{O}(K)$. Moreover, the algorithm partially accelerates the convergence rate from $\mathcal{O}(1/\tau)$ to $\mathcal{O}(1/\tau^2)$. This acceleration makes the approach particularly attractive, as it allows the use of larger Lipschitz constants L_p and L_δ [24].

We now present the proposed FPLA-PSH algorithm in detail as Algorithm 1. In particular, Lines 3-7 perform a parallelized update across the variable blocks p and δ to accelerate convergence, Line 8 updates the probabilistic scheduling variable $\nu(t)$, Lines 9 and 10 update the dual variables, and Lines 11 and 12 update the weight parameters. The algorithm terminates when the change in the solution $p(t)$ falls below a prescribed tolerance ε, and outputs the optimized solution $\hat{p}$.

Algorithm 1 FPLA-PSH Algorithm in Single-Objective Cases

Input: Setting $(N,\ K,\ P,\ T,\ B,\ \sigma^2,\ (\rho,\ a,\ b,\ A))$, user set $\mathcal{K}$, channels $\{\boldsymbol{h}_k\}$, channels gain matrix $\boldsymbol{G}$, gain diagonal matrix $\boldsymbol{D}$, $\lambda_p \geq 2K(\|\boldsymbol{G}-\boldsymbol{D}\|_2+1)^2$, the error tolerance ε, $[\boldsymbol{p}^T,\ \boldsymbol{\delta}^T]^T \triangleq [\boldsymbol{x}_p^T,\ \boldsymbol{x}_\delta^T]^T$ and $\mathcal{M} \triangleq \{p,\ \delta\}$.

Output: The optimized solution $\hat{\boldsymbol{p}}$.

Initialize $t=0,\ \boldsymbol{x}_p(0)=\boldsymbol{z}_p(0)=P/K\times\boldsymbol{1},\ \boldsymbol{x}_\delta(0)=\boldsymbol{z}_\delta(0)=(\boldsymbol{G}-\boldsymbol{D})\boldsymbol{x}_p(0)+\sigma^2\boldsymbol{1},\ \boldsymbol{\alpha}(0)=1/K\times\boldsymbol{1},\ \beta(0)=1,\ \mu(0)=\theta(0)=1,\ \nu(0)=0$;
repeat
  for $b\in\mathcal{M}$ in parallel **do**
    Update $\boldsymbol{y}_b(t+1)$;
    Update $\boldsymbol{z}_b(t+1)$;
    Update $\boldsymbol{x}_b(t+1)$;
  end for
  Update $\nu(t+1)$ as per (13);
  Update $\boldsymbol{\alpha}(t+1)$ as per (24a);
  Update $\beta(t+1)$ as per (24b);
  Update $\theta(t+1)$ as per (17c);
  Update $\mu(t+1)=1/\theta(t+1)$;
  $t=t+1$;
until $\|\boldsymbol{p}(t)-\boldsymbol{p}(t-1)\|_2\leq\varepsilon$;
$\hat{\boldsymbol{p}}=\boldsymbol{x}_p(t)$.

IV. TASKS SCHEDULING CASE: PARALLEL ALGORITHM

We have solved users scheduling optimization problem. However, the coupling between users and tasks scheduling impedes the learning performance, thus we derive an asymptotic case in the absence of CCI to further get deeper insights on the tasks scheduling cases [25], [26]. Specifically, the interference amplification, i.e., $G_{k,\ell}\to 0 (k\neq\ell)$, tends to infinity almost surely via the law of large numbers when the number of antennas N approaches to infinite ($N\to\infty$). Then, we simplify the user data-rate as

$$R_k=\log_2\left(1+\frac{G_{k,k}p_k}{\sigma^2}\right),$$

and give the number of collected samples as

$$D_i^r=\sum_{k\in\mathcal{K}_i}\frac{BTR_k}{V_i}=\sum_{k\in\mathcal{K}_i}\frac{BT}{V_i}\log_2\left(1+\frac{G_{k,k}p_k}{\sigma^2}\right). \quad (25)$$

Here, we eliminate probabilistic scheduling since the key characteristic of this specific scenario is that the number of antennas is much greater than the number of users, i.e., $K\ll N$. This is a small-scale users' case, and the co-channels maintain orthogonality, eliminating the interference influences. Putting (25) into $\mathcal{P}3$ yields

$$\mathcal{P}4:\min_{\boldsymbol{p}}\ \max_{i\in\mathcal{I}}\ \Phi_i(\boldsymbol{p}_i|\sigma^2)$$
$$\text{s.t.}\ \sum_{k\in\mathcal{K}}p_k=P, \quad (26a)$$

where $\boldsymbol{p}_i\triangleq[p_1,\ p_2,\cdots,\ p_{|\mathcal{K}_i|}]^T$ and $\boldsymbol{p}\triangleq[\boldsymbol{p}_1^T,\ \boldsymbol{p}_2^T,\cdots,\ \boldsymbol{p}_I^T]^T$ with

$$\Phi_i(\boldsymbol{p}_i|\sigma^2)\triangleq\rho_i\times a_i\left(\sum_{k\in\mathcal{K}_i}\frac{BT}{V_i}\log_2\left(1+\frac{G_{k,k}p_k}{\sigma^2}\right)+A_i\right)^{-b_i}.$$

It is obvious that how multi-objective learning tasks influence resource allocation efficiently.

We next introduce additional variables P_i to decompose the power constraint (26a), and obtain a set of sub-problems in parallel. Accordingly, $\mathcal{P}4$ is transformed as

$$\mathcal{P}5:\min_{\{\boldsymbol{p}_i\}_{i=1}^I,\{P_i\}_{i=1}^I,u}\ u+\frac{1}{2c}(u-u_m)^2$$
$$\text{s.t.}\ \omega_i\Phi_i(\boldsymbol{p}_i|\sigma^2)\leq u,\ \omega_i\in\{0,\ 1\}, \quad (27a)$$
$$\sum_{k\in\mathcal{K}_i}p_k=P_i,\ p_k\geq 0,\ \forall i\in\mathcal{I}, \quad (27b)$$
$$\sum_{i\in\mathcal{I}}P_i=P, \quad (27c)$$

where $u_m\triangleq\max_{i\in\mathcal{I}}\Phi_i(\boldsymbol{p}_i|\sigma^2)$ denotes the worst learning error, c is a penalty parameter, $u\in[0,1]$ is a slack variable and has the interpretation of classification error level, (27b) and (27c) come from (26a), which is divided into I blocks in parallel [13]. The objective function of $\mathcal{P}5$ is the epigraph function with a penalty, constraint (27a) denotes the epigraph form of the objective function in $\mathcal{P}4$, here $u_m\leq u$.

Using a similar approach to solve $\mathcal{P}2$, the ALF of $\mathcal{P}5$ is given by

$$L(\{\boldsymbol{p}_i\}_{i=1}^I,\{P_i\}_{i=1}^I,u;\{\lambda_i\}_{i=1}^I,\{\beta_i\}_{i=1}^I)$$
$$=u+\frac{1}{2c}(u-u_m)^2+\sum_{i\in\mathcal{I}}\lambda_i\left(\Phi_i(\boldsymbol{p}_i|\sigma^2)-u\right)$$
$$+\sum_{i\in\mathcal{I}}\beta_i\left(\sum_{k\in\mathcal{K}_i}p_k-P_i\right)+\frac{\mu}{2}\sum_{i\in\mathcal{I}}\left(\sum_{k\in\mathcal{K}_i}p_k-P_i\right)^2.$$

It is hard to update $\{P_i\}_{i=1,\cdots,I}$ since the resource optimization for various learning tasks exhibits coupling. Therefore, we derive Proposition 3 to address this issue.

Proposition 3 (Parallelization [13], [14]): Given ALF, we can make the following iteration concerning variables P_i:

$$P_i(t)\triangleq\boldsymbol{1}^T\boldsymbol{p}_i(t)-\frac{1}{I}\left(\boldsymbol{1}^T\boldsymbol{p}(t)-P\right). \quad (28)$$

The relative dual variables $\{\beta_i\}_{i=1}^I$ are updated by

$$\beta(t+1)=\max\left(\beta(t)+\frac{\mu}{I}\left(\boldsymbol{1}^T\boldsymbol{p}(t+1)-P\right),\ 0\right), \quad (29)$$

and $\beta_i(t+1)=\beta(t+1)$, for all $i=1,\cdots,I$.

Proposition 3 effectively develops an efficient parameter update metric in parallel. Proposition 3 also shows that we

can eliminate (27c) by (28), efficiently. Thus, the ALF of $\mathcal{P}2$ can be divided into a set of sub-functions, given by

$$L_i\left(\boldsymbol{p}_i;\lambda_i,\beta\right)=\lambda_i\left(\Phi_i(\boldsymbol{p}_i|\sigma^2)-u\right)+\beta\left(\sum_{k\in\mathcal{K}_i}p_k-P_i(t)\right)+\frac{\mu}{2}\left(\sum_{k\in\mathcal{K}_i}p_k-P_i(t)\right)^2.$$

We next compute partial derivatives of $L_i\left(\boldsymbol{p}_i;\lambda_i,\beta\right)$ with respect to relative variables, and then apply the gradient descent method in parallel.

A. UPDATE p_i WITH OTHER VARIABLES FIXED IN PARALLEL

It can be seen that $L_i\left(\boldsymbol{p}_i;\lambda_i,\beta\right)$ is differentiable with respect to $\boldsymbol{p}_i$, and the corresponding gradients are

$$\nabla_{\boldsymbol{p}_i}L_i\left(\boldsymbol{p}_i;\lambda_i,\beta\right)=\left[\frac{\partial L_1}{\partial p_1},\frac{\partial L_2}{\partial p_2},\cdots,\frac{\partial L_{|\mathcal{K}_i|}}{\partial p_{|\mathcal{K}_i|}}\right]^T,$$

with its k^{th} element being

$$\frac{\partial L_i}{\partial p_k}=\left(-b_i\lambda_i\rho_i a_i\left(\sum_{k\in\mathcal{K}_i}\frac{BT}{V_i}\log_2\left(1+\frac{G_{k,k}p_k}{\sigma^2}\right)+A_i\right)^{-b_i-1}\frac{BTG_{k,k}}{\ln(2)V_i(\sigma^2+G_{k,k}p_k)}\right)+\beta+\mu\left(\sum_{k\in\mathcal{K}_i}p_k-P_i(t)\right),$$

then we apply a gradient descent method to obtain

$$\boldsymbol{p}_i(t+1)=\max\left(\boldsymbol{p}_i(t)-\eta\nabla_{\boldsymbol{p}_i}L_i\left(\boldsymbol{p}_i(t);\lambda_i(t),\beta(t)\right),\ \mathbf{0}\right). \tag{30}$$

Proposition 3 shows that the term $\sum_{k\in\mathcal{K}_i}p_k(t)-P_i(t)$ is replaced by $\Delta(t)$, which is independent of additional variables $P_i(t)$. Moreover, it implies that Proposition 3 can speed up the convergence of our algorithm.

B. UPDATE RELATIVE DUAL VARIABLES

We can obtain an optimized solution directly, given by

$$\lambda_i(t+1)=\min\left(\max\left(\lambda_i(t)+\eta\left(\Phi_i(\boldsymbol{p}_i(t+1)|\sigma^2)-u(t)\right),\ 0\right),\ 1\right). \tag{31}$$

And, the dual variable $\beta(t+1)$ can be updated by (29).

C. UPDATE u WITH OTHER VARIABLES FIXED

By use of gradient descent method, we obtain

$$u(t+1)=u(t)-\eta\left(1-\sum_{i\in\mathcal{I}}\lambda_i(t+1)+\frac{u(t)-u_m}{c}\right). \tag{32}$$

Algorithm 2 Parallel Algorithm for Multi-Objective Learning Tasks

Input: Setting $(I,N,K,P,T,B,\sigma^2,A_i)$, channels $\{\boldsymbol{h}_k\}$, sets $\{\mathcal{K}_i\}$ for $i\in\mathcal{I}$, gain diagonal matrix $\boldsymbol{D}$, learning rate η and the error tolerance ε.

Output: The optimized solution $\hat{\boldsymbol{p}}$.

1: Initialize $t=0$, $\boldsymbol{p}(0)=\mathbf{1}$, $\lambda(0)=1$, $\beta(0)=1$, $u(0)=1$;
2: **repeat**
3: **for** $i\in\mathcal{I}$ in parallel **do**
4: Update $\boldsymbol{p}_i(t+1)$ as per (30);
5: Update $\lambda_i(t+1)$ as per (31);
6: **end for**
7: Update $\beta(t+1)$ as per (29);
8: Update $u(t+1)$ as per (32);
9: $t=t+1$;
10: **until** $\|\boldsymbol{p}(t)-\boldsymbol{p}(t-1)\|_\infty\le\varepsilon$.
11: $\hat{\boldsymbol{p}}=\boldsymbol{p}(t)$.

TABLE 2. The simulated parameters' setting [12], [13], [25].

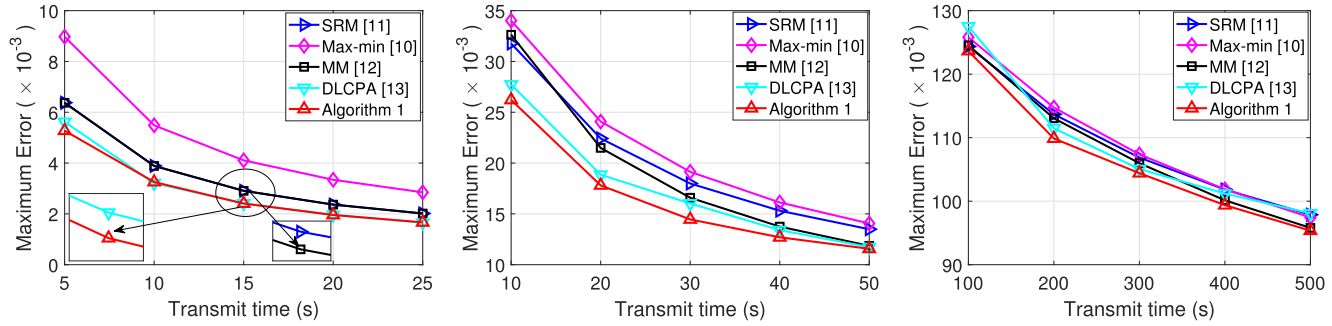
Parameters	Values	Parameters	Values
Noise power	$\sigma^2=-77\,\text{dBm}$	Transmit power	$P=13\,\text{dBm}$
Communication bandwidth	$B=180\,\text{kHz}$	Path loss	$\varrho_k=-90\,\text{dB}$
The number of users	$\lvert\mathcal{K}_1\rvert=\cdots=\lvert\mathcal{K}_I\rvert=40$	Number of antennas	$N=5$
Models	**Datasets**	**Symbols**	**Values**
SVM [19]	Digits	(a_1,b_1,A_1,V_1)	(5.2, 0.72, 200, 324 bits)
CNN6 [18]	MNIST	(a_2,b_2,A_2,V_2)	(7.3, 0.69, 300, 6276 bits)
ResNet110 [20]	CIFAR10	(a_3,b_3,A_3,V_3)	(8.15, 0.44, 1600, 24584 bits)
PointNet [12]	ModelNet40	(a_4,b_4,A_4,V_4)	(0.96, 0.24, 800, 192008 bits)

It is evident that the selection of a learning-oriented objective, depending on task difficulty, improves resource utilization efficiency at the edge, which increases the achievable learning gain. In terms of computational complexity, it is proportional to K primal variables, $I+1$ dual variables, and one auxiliary variable. Here, K primal and I dual variables can be in parallel due to multi-objective optimizations. Moreover, parallel acceleration is to speed up the algorithm further. Consequently, the total complexity is $\mathcal{O}(\tau\ (K/I+2))$, due to I parallel tasks. Now, we show Algorithm 2 in detail. Specifically, Lines 3-6 incorporate task-based parallel modules, and Lines 7-8 describe information aggregation modules.

V. SIMULATION RESULTS AND DISCUSSIONS

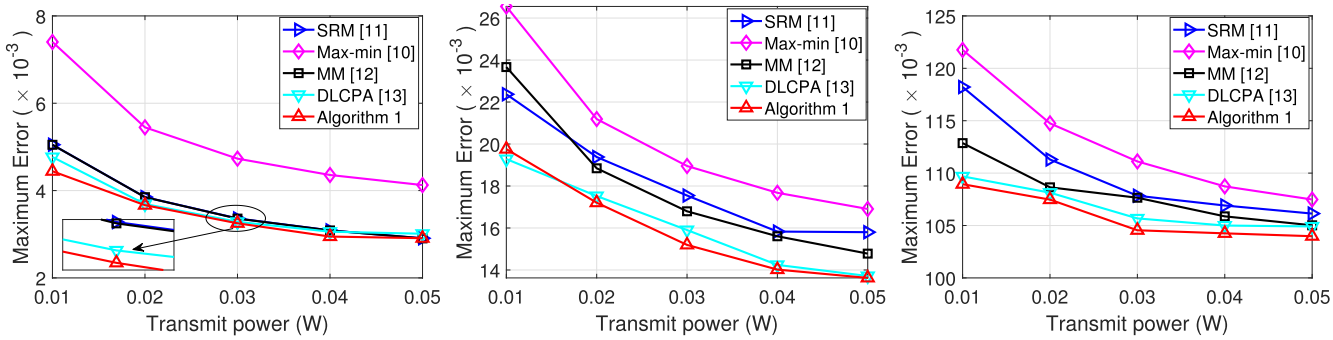
This section provides simulation results to evaluate the proposed algorithm performances. We set the simulated parameters, as shown in Table 2 unless specified otherwise. In particular, Rows 1-4 and Rows 5-9 represent the wireless and learning parameters, respectively. The transmit times T are set as $10\,s$, $20\,s$, and $200\,s$ for three different simulated cases: single-task (i.e., SVM), two-task (i.e., SVM and CNN6) and four-task (i.e., SVM, CNN6, ResNet110, and PointNet) cases, respectively.

In our simulations, we employ the workstation with the Hygon C68-3250 CPU and 8-core processors configuration and use Matlab to conduct 100 Monte Carlo simulations.



(a) Single-task (b) Two tasks (c) Four-tasks

FIGURE 2. The maximum error of different learning tasks versus the total transmission time T.



(a) Single-task, $T = 10s$. (b) Two tasks, $T = 20s$. (c) Four tasks, $T = 200s$.

FIGURE 3. Maximum error of different learning tasks versus transmit power $P(W)$.

Moreover, we also assign threads according to the number of learning tasks for parallelization implementation.

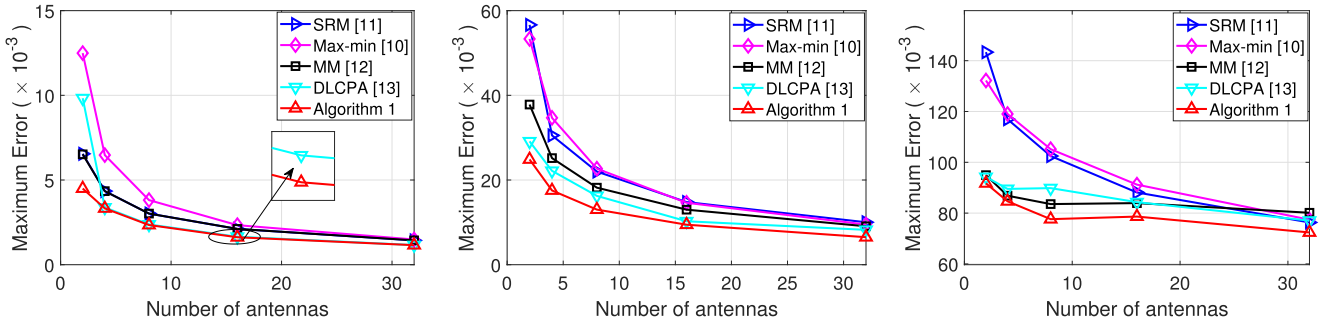
A. USERS SCHEDULING PERFORMANCE

This subsection presents the relationships between the learning and resource allocation in different algorithms. Besides our proposed learning-oriented algorithms (i.e., Algorithm 1), we also simulate other benchmark schemes: 1) Max-min fairness scheme (Max-min for short) [10]; 2) Sum-rate maximization scheme (SRM for short) [11]; 3) MM-based LCPA (MM for short) scheme [12]; 4) Distributed LCPA scheme (DLCPA for short) [13].

Figure 2 depicts the maximum errors versus the total transmission time. Specifically, Fig. 2(a) shows that the learning error of the proposed algorithm gradually decreases with increasing transmit time and consistently remains below other curves. It benefits from probabilistic scheduling algorithms, which decreases the influence of co-channel interference, then obtains more training data under joint communication. Compared with [13], the proposed model design a probabilistic scheduling metric, enabling the stochastic channel distribution. Moreover, Figs. 2(b)-(c) also achieves comparable performance in two-task and four-task cases, respectively. Beyond the scheduling algorithm, it also benefits from that the designed algorithm is adaptive to the min-max learning model, ensuing the optimized model selection.

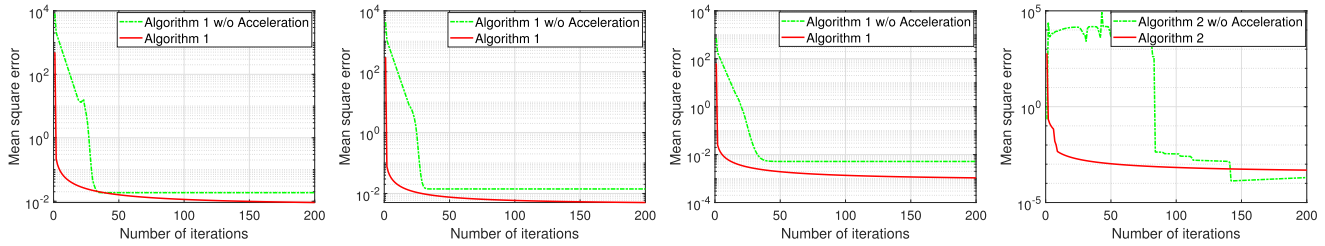
Figure 3 shows the maximum errors versus the transmit power. Specifically, Fig. 3(a) demonstrates a monotonically decreasing trend of learning errors, while the proposed algorithm also maintains a superiority among all benchmark curves. This improvement mainly stems from enhanced data collection with higher power budgets. Moreover, the proposed model adopts a learning-oriented communication scheme and allocates data-collection resources according to the selected worst-case learning task. Similarly, Figs. 3(b)-(c) also exhibit consistent trends in two-task and four-task cases, respectively. It also benefits from probabilistic scheduling and learning-oriented communication.

Figure 4 describes the maximum errors versus the number of antennas. Specifically, Fig. 4(a) shows a monotonically decreasing trend of learning errors. Moreover, the gap of the maximum errors between the proposed model and other models progressively narrows with increasing number of antennas. It is because that the co-channel interference decreases when the number of antennas increases gradually and therefore leads to a good learning performance. Similarly, Figs. 4(b)-(c) also exhibit consistent trends in two-task and four-task cases, respectively. It is mainly because that the proposed model adaptively optimizes for



(a) Single-task, $T = 10s$. (b) Two tasks, $T = 20s$. (c) Four tasks, $T = 200s$.

FIGURE 4. Maximum error of different learning tasks versus the number of antennas N.



(a) The $N = 2$ case. (b) The $N = 16$ case. (c) The $N = 64$ case. (d) The asymptotic case.

FIGURE 5. Mean squared errors vs. the number of iterations in four tasks cases, with $T = 200$ s and the learning step $\eta = 1 \times 10^{-3}$.

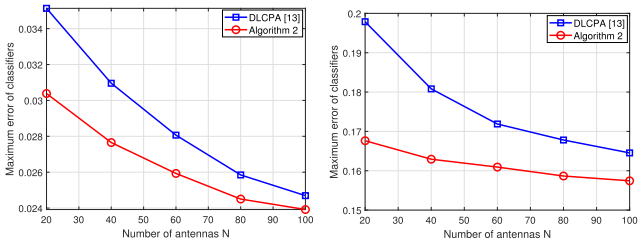
the worst-case case, leading to a marginal performance gain.

B. TASKS SCHEDULING PERFORMANCE

This subsection mainly describes the relationships between the learning and differ learning tasks in the asymptotic case. The number of users is set as $|\mathcal{K}_1| = \cdots = |\mathcal{K}_I| = 2$. To evaluate the convergence process, we define a mean squared error (MSE) as

$$\begin{aligned}\text{MSE} &\triangleq \|\boldsymbol{p}(t) - \boldsymbol{p}(t-1)\|_2 + \|\boldsymbol{\delta}(t) - \boldsymbol{\delta}(t-1)\|_2 + |\|\boldsymbol{p}(t)\|_1 - P| \\ &\quad + \|(\boldsymbol{G} - \boldsymbol{D})\boldsymbol{p}(t) - \boldsymbol{\delta}(t) + \sigma^2\mathbf{1}\|_2. \end{aligned} \tag{33}$$

Figure 5 depicts the MSE computed by (33) versus the number of iterations. On the one hand, we observe from Fig. 5(a) that Algorithm 1 with acceleration outperforms that without it in terms of convergence speed. The reason is due to the accelerated convergence rate ($\mathcal{O}(1/\tau^2)$) for the proposed algorithm. Similarly, Figs. 5(b)-5(d) illustrate that the accelerated algorithm also benefits from faster convergence. On the other hand, Fig. 5(d) shows that the performance of Algorithm 2 suffers from a slower mean squared error than Algorithm 1. Moreover, Figs. 5(a)-5(c) show that when Algorithm 1 reaches convergence, its mean squared error decreases as the number of antennas increases. The reason behind these observations is that the influence of interference diminishes with an increasing number of antennas. Therefore, the optimization problem transforms from nonconvex

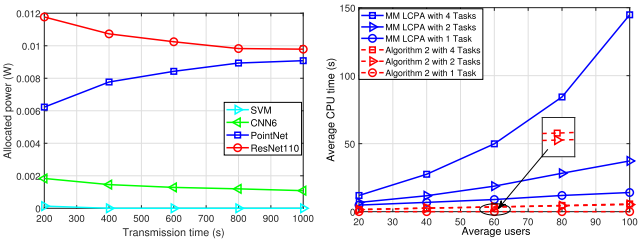


(a) Two tasks, $T = 20$ s. (b) Four tasks, $T = 200$ s.

FIGURE 6. The learning performance in the asymptotic case.

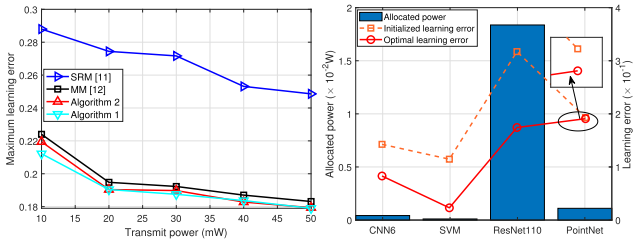
$\mathcal{P}1$ to less nonconvex $\mathcal{P}4$, making Algorithm 1 achieve near-optimal solutions with high probability. Specifically, when the number of antennas N approaches to infinite, Algorithm 1 converges to a closed-form optimal solution.

To get deeper insights on the learning performance, Fig. 6 describes the maximum errors versus the number of antennas in the asymptotic case. Specifically, Fig. 6(a) shows that the curve of Algorithm 2 is consistently below one of DLCPA algorithms in the two-task case. It is because that the interference of Algorithm 2 is close to 0, resulting in a higher learning gain than DLCPA algorithms. Moreover, when the number of antennas increases, the gap between Algorithm 2 and DLCPA algorithms gradually decreases. The reason is that the channel interference decreases when the number of antennas increases. Comparing with Fig. 6(a), Fig. 6(b) also observes the similar performance in the four-task case.



(a) Average transmit power. (b) Average CPU time.

FIGURE 7. Comparison of average transmit power and average CPU time.



(a) Maximum learning error versus transmit power. (b) The allocation of transmit power.

FIGURE 8. Comparison during different tasks.

Moreover, the gap in Fig. 6(b) is larger than one of Fig. 6(a). It is because that Fig. 6(b) has more users and communication channels, resulting in a higher interference than Fig. 6(a).

Figure 7 presents the performance of the power allocation scheme and CPU running time in the asymptotic case, respectively. On the one hand, Fig. 7(a) shows that allocating much more power resources to the ResNet110 and PointNet tasks than other two tasks, because there are usually millions of parameter between these two models and training them is more difficult, especially for ResNet110. Moreover, the optimization process shifts the point from ResNet110 to PointNet as transmission time increases, since this method reduces the error rate of ResNet110 drastically with increasing sample size, the error of PointNet exceeds that of ResNet110. However, the CNN6 and SVM parameter scale is relatively small, thus power allocation among these two models change little with transmission time, especially for SVM. On the other hand, Fig. 7(b) shows that Algorithm 2 for the single-task case has a shorter CPU time than the MM algorithm developed in [12]. The reason is due to the low computational complexity ($\mathcal{O}(K)$) for the proposed algorithm. Moreover, Fig. 7(b) also shows that the CPU time of Algorithm 2 remains almost the same as the number of tasks from two to four cases, compared with the single task case. The reason behind this observation is that the per-iteration complexity of our parallel algorithm is almost $\mathcal{O}(K)$ for each task.

Figure 8 shows the performance of the transmit power in the four-task case. On the one hand, Fig. 8(a) presents that MLE decrease gradually as the transmit power increases. It is because that the number of collected training samples increases with a larger allocated power. Moreover, our algorithms maintain the lower MLE than other algorithms. It benefits from the probabilistic scheduling algorithm. Therefore, we conclude that our probabilistic scheduling algorithms improve the efficiency of the transmit power. On the other hand, Fig. 8(b) shows that our algorithms allocate almost the power to ResNet110 during optimization. It is because that ResNet110 has more network parameters than other tasks and requires more power to transmit data for training. When the learning error of ResNet110 is lower than that of PointNet, the system allocates some power to PointNet. Although a fraction of power is allocated to CNN6 and SVM, the learning error decreases significantly compared with PointNet. Here, the training data of PointNet is a 3D point clouds dataset, which is large and difficult to be transmitted, thus the decline of learning error is not obvious for PointNet.

VI. CONCLUSION

The learning-aware optimization model with probabilistic scheduling has been proposed to balance learning performance and resource allocation. First, the users scheduling algorithm has been developed to reveal the relationship between resource allocation and learning performance. Moreover, a fast proximal linearized algorithm is designed to solve the large-scale optimization problems. Then, tasks scheduling in asymptotic cases is derived to improve resource utilization efficiency. Extensive experimental results have shown that the probabilistic scheduling could efficiently improve learning gains and resource utilization ratios. The fast and parallel algorithms enabled large-scale users and different learning tasks efficiently.

APPENDIX A
PROOF OF THEOREM 1

From the t^{th} to the $(t+1)^{th}$ iteration, the expected increment in the dual function in (A.1a) and (A.1b), as shown at the bottom of the next page, where $\Phi^*(\cdot)$ is dual functions of $\Phi(\cdot)$, $\boldsymbol{u}_p, \boldsymbol{u}_\delta$ are relative dual variables of $\boldsymbol{p}$, $\boldsymbol{\delta}$, and $v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) = \left[\boldsymbol{p}(t)^T, \boldsymbol{\delta}(t)^T\right]\left[\boldsymbol{u}_p(t)^T, \boldsymbol{u}_\delta(t)^T\right]^T$ denotes the global change to be kept consistent. Moreover, (A.1a) follows Assumption 3 [17] and

$$\Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) = \sum_{k\in\mathcal{K}} \Phi^*(\mathbf{0}; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t)).$$

Since $\Phi(\cdot)$ is L_m-smooth, $\Phi^*(\cdot)$ is $1/L_m$-strongly convex. Hence, there exist a scalar $\beta \in (0, 1)$, such that

$$\Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) - \sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u^*_{k,p}, \Delta u^*_{k,\delta}; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t)) \tag{A.2a}$$

$$\leq \beta\left[\Phi^*(\boldsymbol{u}^*_p, \boldsymbol{u}^*_\delta) - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t))\right], \tag{A.2b}$$

where $\beta \in (0,1)$, and $v(\boldsymbol{u}_p, \boldsymbol{u}_\delta) = \boldsymbol{p}^H \boldsymbol{u}_p + \boldsymbol{\delta}^H \boldsymbol{u}_\delta$. As such, we have the following:

$$\begin{aligned}
&\Phi^*(\boldsymbol{u}_p^*, \boldsymbol{u}_\delta^*) - \Phi^*(\boldsymbol{u}_p(t+1), \boldsymbol{u}_\delta(t+1)) \\
&\le \Phi^*(\boldsymbol{u}_p^*, \boldsymbol{u}_\delta^*) - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) \\
&\quad + (1-\beta)\mathcal{U}_k \Big[\Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) \\
&\quad - \sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u_{k,p}^*, \Delta u_{k,\delta}^*; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t))\Big] \\
&\le [1-(1-\beta)\mathcal{U}_k]\, \Phi^*(\boldsymbol{u}_p^*, \boldsymbol{u}_\delta^*) - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) \quad \text{(A.3a)} \\
&\le [1-(1-\beta)\mathcal{U}_k]^t\, \Phi^*(\boldsymbol{u}_p^*, \boldsymbol{u}_\delta^*) - \Phi^*(\boldsymbol{u}_p(0), \boldsymbol{u}_\delta(0)). \quad \text{(A.3b)}
\end{aligned}$$

The result then follows by upper bounding R.H.S of (A.3b).

APPENDIX B
PROOF OF PROPOSITION 1

From Eq. (13), we obtain the following expression.

$$(\ell+1)v(\ell+1) - \ell v(\ell) = \frac{\sum_{k=1}^K \mathbb{1}\{w_k(\ell)=1,\ R_k(\ell)>\gamma\}}{K}.$$

Summing above equation from $\ell = 0$ to $\ell = t-1$ and performing some algebraic manipulation yields $v(t)$ of (10).

To proof the steady-state expectation, we first define $v'(\ell)$ as

$$v'(\ell) \triangleq \frac{1}{K}\sum_{k=1}^K \mathbb{1}\{w_k(\ell)=1, R_k(\ell)>\gamma\}.$$

Substituting $v'(\ell)$ into (10) yields:

$$\mathcal{U}_k = \lim_{t\to\infty}\frac{1}{Kt}\sum_{\ell=0}^{t-1}\big(K\cdot v'(\ell)\big) = \lim_{t\to\infty}\underbrace{\frac{1}{t}\sum_{\ell=0}^{t-1} v'(\ell)}_{v(t)}. \quad \text{(B.1)}$$

The final expression, $\lim_{t\to\infty}\frac{1}{t}\sum_{\ell=0}^{t-1} v'(\ell)$, is precisely the long-term *time average* of the process $v(t)$.

By Assumption 1, $v(t)$ is a stationary and ergodic process. For such a process, the time average converges almost surely (and thus in the sense of the limit shown) to the ensemble expectation. Hence,

$$\lim_{t\to\infty}\frac{1}{t}\sum_{\ell=0}^{t-1} v'(\ell) = \mathbb{E}[v(t)],$$

where the expectation $\mathbb{E}[v(t)]$ is constant over time due to stationarity.

Therefore, we have established the equality:

$$\mathcal{U}_k = \lim_{t\to\infty}\frac{1}{t}\sum_{\ell=0}^{t-1} v'(\ell) = \mathbb{E}[v(t)],$$

which completes the proof, confirming the equivalence between (10) and (14).

APPENDIX C
PROOF OF PROPOSITION 2

It can be seen that minimizing $\mathcal{P}3$ is obtained with the identifications

$$\boldsymbol{A} \triangleq \begin{bmatrix} \tilde{\boldsymbol{G}} - \tilde{\boldsymbol{D}} & -\boldsymbol{I} \\ \boldsymbol{1}^T & \boldsymbol{0}^T \end{bmatrix},\ \boldsymbol{A}_1 \triangleq \begin{bmatrix} \tilde{\boldsymbol{G}} - \tilde{\boldsymbol{D}} \\ \boldsymbol{1}^T \end{bmatrix},\ \boldsymbol{A}_2 \triangleq \begin{bmatrix} -\boldsymbol{I} \\ \boldsymbol{0}^T \end{bmatrix},$$

where $\tilde{\boldsymbol{G}} \triangleq \begin{bmatrix} \boldsymbol{G}(1,:)^T \bullet \boldsymbol{w}(t) & \cdots & \boldsymbol{G}(K,:)^T \bullet \boldsymbol{w}(t) \end{bmatrix}^T$ and $\tilde{\boldsymbol{D}} \triangleq \begin{bmatrix} \boldsymbol{D}(:,1) \bullet \boldsymbol{w}(t) & \cdots & \boldsymbol{D}(:,K) \bullet \boldsymbol{w}(t) \end{bmatrix}$. So (5a) and (9a) can be equivalent to the following equation.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{r},$$

where $\boldsymbol{r} \triangleq \begin{bmatrix} -\sigma^2 \boldsymbol{1}^T & P \end{bmatrix}^T$ and $\boldsymbol{x} \triangleq \begin{bmatrix} \boldsymbol{p}^T & \boldsymbol{\delta}^T \end{bmatrix}^T$. Moreover, the ALF of $\mathcal{P}2$ can be rewritten as follows:

$$f(\boldsymbol{x}) \triangleq \Phi(\boldsymbol{x}) + \langle \lambda, \boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}\rangle + \frac{\mu}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{r}\|_2^2,$$

where $\lambda \triangleq \begin{bmatrix} \boldsymbol{\alpha}^T & \beta \end{bmatrix}^T$ and $\Phi(\boldsymbol{x}) \triangleq \Phi(\boldsymbol{p}|\boldsymbol{\delta})$.

Next, we consider $f(\boldsymbol{x})$ with $n = 2$ blocks of variables. The state-of-the-art solver for $\mathcal{P}2$ is a proximal fast linearized ADMM with parallel splitting which updates each block [24]. We start with writing one of

$$\begin{aligned}
&\Phi^*(\boldsymbol{u}_p(t+1), \boldsymbol{u}_\delta(t+1)) - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) \\
&\ge \mathcal{U}_k \Bigg[\sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u_{k,p}^*, \Delta u_{k,\delta}^*; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t)) \\
&\quad + \sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u_{k,p}(t), \Delta u_{k,\delta}(t); v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t)) \\
&\quad - \sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u_{k,p}^*, \Delta u_{k,\delta}^*; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t))\Bigg] \quad \text{(A.1a)} \\
&\ge (1-\beta)\mathcal{U}_k \Bigg[\sum_{k\in\mathcal{K}} \Delta\Phi^*(\Delta u_{k,p}^*, \Delta u_{k,\delta}^*; v(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)), u_{k,p}(t), u_{k,\delta}(t)) - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t))\Bigg] \\
&\quad - \Phi^*(\boldsymbol{u}_p(t), \boldsymbol{u}_\delta(t)) \quad \text{(A.1b)}
\end{aligned}$$

ALFs $L\left(\boldsymbol{p}|\nabla_{\boldsymbol{y}_p}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)),\boldsymbol{p}(t),z_p(t),\boldsymbol{\delta}(t);\boldsymbol{\alpha}(t),\beta(t)\right)$ as follows:

$$
\begin{aligned}
&L\left(\boldsymbol{p}|\nabla_{\boldsymbol{y}_p}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)),\boldsymbol{p}(t),z_p(t),\boldsymbol{\delta}(t);\boldsymbol{\alpha}(t),\beta(t)\right)\\
&=\langle\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)),\boldsymbol{p}\rangle+\mu(t)\langle\boldsymbol{A}_1^T(\boldsymbol{A}\boldsymbol{z}_x(t)-\boldsymbol{r}),\boldsymbol{p}\rangle\\
&\quad+\frac{L_p\theta(t)+\mu(t)\lambda_p}{2}\|\boldsymbol{p}-z_p(t)\|_2^2+\langle\lambda(t),\boldsymbol{A}_1\boldsymbol{p}\rangle\\
&=\langle\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{y}_p(t+1)|\boldsymbol{\delta}(t)),\boldsymbol{p}\rangle+\sum_{k\in\mathcal{K}}\alpha_k(t)\sum_{\ell\in\mathcal{K}\backslash k}w_\ell(t)G_{k,\ell}p_\ell\\
&\quad+\mu(t)\left\langle\left(\sum_{k\in\mathcal{K}}z_{p_k}(t)-P\right)\mathbf{1},\boldsymbol{p}\right\rangle+\beta(t)\sum_{k\in\mathcal{K}}p_k\\
&\quad+\mu(t)\Big\langle(\boldsymbol{G}-\boldsymbol{D})^T\left((\boldsymbol{G}-\boldsymbol{D})(\boldsymbol{w}(t)\bullet z_p(t))-z_\delta(t)+\sigma^2\mathbf{1}\right),\\
&\quad\boldsymbol{w}(t)\bullet\boldsymbol{p}\rangle+\frac{L_p\theta(t)\mu(t)\lambda_p}{2}\left\|\boldsymbol{p}-z_p(t)\right\|_2^2
\end{aligned}
\tag{C.1}
$$

where L_p is Lipschitz constant of $\nabla_{\boldsymbol{p}}\Phi(\boldsymbol{p}|\boldsymbol{\delta}(t))$ in Lemma 2, $z_x\triangleq\left[\begin{array}{cc}z_p^T & z_\delta^T\end{array}\right]^T$ and $\boldsymbol{y}_p(t+1)$ is updated by (17a) to accelerate $\Phi(\boldsymbol{p}|\boldsymbol{\delta}(t))$. Here, $\lambda_p\geq 2\|\boldsymbol{A}_1\|_2^2$ should be satisfied, so we can obtain

$$
\begin{aligned}
&\lambda_p\geq 2K\left(\|\boldsymbol{G}-\boldsymbol{D}\|_2+1\right)^2\geq 2\left(\|\boldsymbol{w}(t)\|_2\|\boldsymbol{G}-\boldsymbol{D}\|_2+\sqrt{K}\right)^2\\
&\geq 2\left(\|\tilde{\boldsymbol{G}}-\tilde{\boldsymbol{D}}\|_2+\|\mathbf{1}\|_2\right)^2\geq 2\|\boldsymbol{A}_1\|_2^2.
\end{aligned}
$$

Then, we also write another ALF $L(\boldsymbol{\delta}|\nabla_{\boldsymbol{y}_\delta}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)),\boldsymbol{p}(t+1),\boldsymbol{\delta}(t),z_\delta(t);\boldsymbol{\alpha}(t),\beta(t))$ as follows:

$$
\begin{aligned}
&L\left(\boldsymbol{\delta}|\nabla_{\boldsymbol{y}_\delta}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)),\boldsymbol{p}(t+1),\boldsymbol{\delta}(t),z_\delta(t);\boldsymbol{\alpha}(t),\beta(t)\right)\\
&=\langle\nabla_{\boldsymbol{\delta}}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)),\boldsymbol{\delta}\rangle+\mu(t)\langle\boldsymbol{A}_2^T(\boldsymbol{A}\boldsymbol{z}_x(t)-\boldsymbol{r}),\boldsymbol{\delta}\rangle\\
&\quad+\frac{L_\delta\theta(t)+2\mu(t)}{2}\|\boldsymbol{\delta}-z_\delta(t)\|_2^2+\langle\lambda(t),\boldsymbol{A}_2\boldsymbol{\delta}\rangle\\
&=\langle\nabla_{\boldsymbol{\delta}}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{y}_\delta(t+1)),\boldsymbol{\delta}\rangle-\sum_{k\in\mathcal{K}}\alpha_k(t)\delta_k\\
&\quad+\mu(t)\sum_{k\in\mathcal{K}}\delta_k\left(\sum_{\ell\in\mathcal{K}\backslash k}w_\ell G_{k,\ell}z_{p_\ell}(t)+\sigma^2-z_{\delta_k}(t)\right)\\
&\quad+\frac{L_\delta\theta(t)+2\mu(t)}{2}\|\boldsymbol{\delta}-z_\delta(t)\|_2^2,
\end{aligned}
\tag{C.3}
$$

where L_δ is a Lipschitz constant of $\nabla_{\boldsymbol{\delta}}\Phi(\boldsymbol{p}(t+1)|\boldsymbol{\delta})$ in Lemma 2 and $\boldsymbol{y}_\delta(t+1)$ is updated by (17b) to accelerate $\Phi(\boldsymbol{p}(t+1)|\boldsymbol{\delta})$. Here, it is seen obviously that $\lambda_\delta\geq n\|\boldsymbol{A}_2\|_2^2=2$.

REFERENCES

[1] T. Park and W. Saad, "Distributed learning for low latency machine type communication in a massive Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5562–5576, Jun. 2019.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[3] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[4] A. Mahmoudi, M. Zaher, and E. Björnson, "Low-latency and energy-efficient federated learning over cell-free networks: A trade-off analysis," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 2274–2292, 2025.

[5] H. Lee, S. H. Lee, T. Q. S. Quek, and I. Lee, "Deep learning framework for wireless systems: Applications to optical wireless communications," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 35–41, Mar. 2019.

[6] H. Wu, Z. Zhang, C. Guan, K. Wolter, and M. Xu, "Collaborate edge and cloud computing with distributed deep learning for smart city Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8099–8110, Sep. 2020.

[7] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[8] S. Yu, X. Chen, L. Yang, D. Wu, M. Bennis, and J. Zhang, "Intelligent edge: Leveraging deep imitation learning for mobile edge computation offloading," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 92–99, Feb. 2020.

[9] Y.-C. Liu and C.-T. Chuang, "Combining the improved RGB water-filling algorithm with penumbra removal technique for shadow removal from digitized images," *IEEE Access*, vol. 13, pp. 57847–57857, 2025.

[10] N. Ginige, P. Dharmawansa, A. Sousa De Sena, N. Huda Mahmood, N. Rajatheva, and M. Latva-Aho, "Max-min fairness for stacked intelligent metasurface-assisted multi-user MISO systems," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 5639–5656, 2025.

[11] X. Liu, B. Yang, J. Liu, L. Xian, X. Jiang, and T. Taleb, "Sum-rate maximization for D2D-enabled UAV networks with seamless coverage constraint," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38704–38718, Dec. 2024.

[12] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine intelligence at the edge with learning centric power allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020.

[13] H. Xie, M. Xia, P. Wu, S. Wang, and H. V. Poor, "Edge learning for large-scale Internet of Things with task-oriented efficient communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9517–9532, Dec. 2023.

[14] D. Wen, X. Jiao, P. Liu, G. Zhu, Y. Shi, and K. Huang, "Task-oriented over-the-air computation for multi-device edge AI," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 2039–2053, Mar. 2024.

[15] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.

[16] 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, Jan. 2021.

[17] 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, Jan. 2020.

[18] J. Yang, W. Jiang, and L. Nie, "Hypernetworks-based hierarchical federated learning on hybrid non-IID datasets for digital twin in industrial IoT," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 2, pp. 1413–1423, Mar. 2024.

[19] A. Kurmaiah and C. Vaithilingam, "Optimization of fault identification and location using adaptive neuro-fuzzy inference system and support vector machine for an AC microgrid," *IEEE Access*, vol. 13, pp. 20599–20619, 2025.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[21] X. Chen, J. Chen, S. Qiu, X. Xue, and J. Pu, "GS-net: Point cloud sampling with graph neural networks," *Pattern Recognit.*, vol. 170, Feb. 2026, Art. no. 112054.

[22] E. Harvey, W. Chen, D. M. Kent, and M. C. Hughes, "A probabilistic method to predict classifier accuracy on larger datasets given small pilot data," in *Proc. Mach. Learn. Health (ML4H)*, vol. 225, Dec. 2023, pp. 129–144.

[23] K. Luo et al., "A multi-power law for loss curve prediction across learning rate schedules," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2025, pp. 1–16.

[24] C. Lu, H. Li, Z. Lin, and S. Yan, "Fast proximal linearized alternating direction method of multiplier with parallel splitting," in *Proc. 3rd Conf. Art. Intell. (AAAI)*, Feb. 2016, pp. 739–745.

[25] S. Buzzi, C. D'Andrea, L. Wang, A. Hasim Gokceoglu, and G. Peters, "Co-existing/cooperating multicell massive MIMO and cell-free massive MIMO deployments: Heuristic designs and performance analysis," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 6180–6200, 2024.

[26] H. Zeng, Z. Xie, P. Chen, and Y. Fang, "Expectation propagation detection with physical network coding for massive MIMO systems," *IEEE Signal Process. Lett.*, vol. 31, pp. 41–45, 2024.

HAIHUI XIE received the B.S. and M.S. degrees in photonic and electronic engineering from Fujian Normal University, Fuzhou, China, in 2014 and 2016, respectively, and the Ph.D. degree in information and communication engineering from Sun Yat-sen University, Guangzhou, China, in 2023. He is currently a Lecturer with Fujian Agriculture and Forestry University (FAFU), Fuzhou. His research interests include edge learning, optimization, and wireless communications.

PINGPING XU received the B.S. degree in electronic information engineering from Fujian Agriculture and Forestry University, Fuzhou, China, in 2025, where he is currently pursuing the M.S. degree. His research interests include edge learning and optimization.

ZHAOGANG SHU (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Shantou University, China, in 2002 and 2005, respectively, and the Ph.D. degree from South China University of Technology, Guangzhou, China, in 2008. From September 2008 to July 2012, he was a Senior Engineer and the Project Manager at Ruijie Network Corporation, Fuzhou, China. From October 2018 to October 2019, he was a Visiting Professor with the MOSIAC Laboratory, Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland. He is currently an Associate Professor with the College of Computer and Information Science, Fujian Agriculture and Forestry University, Fuzhou. He is also the Director of the Network System and Cloud Computing Laboratory, Fujian Agriculture and Forestry University. He led more than ten research projects. He was the author of more than 30 articles and five patents. His research interests include software-defined networks, network function virtualization, 5G networks, and next generation network architecture, network security, machine learning-based network optimization, cloud computing, and edge computing. He is a Senior Member of China Computer Federation (CCF) and Fujian Computer Society. He serves as a reviewer for many famous journals on network and communications, including *IEEE Network*, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON NETWORK SERVICE AND MANAGEMENT, *ACM/Springer Mobile Networks*, and *Computer Networks* (Elsevier).

SHUWU CHEN received the M.S. degree in radio physics from Xiamen University, China, in 2003. He is currently a Professor with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University. He is also with the Engineering Research Center of Smart Sensing and Agricultural Chip Technology, Fujian Province University. His research interests include the Internet of Things, edge computing, and AI algorithms.

CHANGSHENG YOU (Member, IEEE) received the B.Eng. degree from the University of Science and Technology of China (USTC), China, in 2014, and the Ph.D. degree from The University of Hong Kong (HKU) in 2018. He was a Research Fellow with the National University of Singapore (NUS). He is currently an Assistant Professor with the Southern University of Science and Technology. His research interests include near-field communications, intelligent reflecting surfaces, UAV communications, edge computing, and learning. He received the IEEE ComSoc Leonard G. Abraham Prize in 2025, the IEEE ComSoc Best Tutorial Paper Award in 2023, the IEEE ComSoc Best Survey Paper Award in 2021, and the IEEE ComSoc Asia–Pacific Region Outstanding Paper Award in 2019. He currently serves an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and IEEE COMMUNICATIONS LETTERS. He is listed as a Clarivate Highly Cited Researcher in 2024–2025 and the IEEE ComSoc Asia-Pacific Best Yong Research in 2024.