RHFedMTL: Resource-Aware Hierarchical Federated Multitask Learning

Xingfu Yi, Rongpeng Li, *Senior Member, IEEE*, Chenghui Peng, Fei Wang, Jianjun Wu, and Zhifeng Zhao, *Member, IEEE*

***Abstract*—The wide applications of artificial intelligence (AI) on massive Internet of Things or smartphones raises significant concerns about privacy, heterogeneity, and resource efficiency. Correspondingly, federated learning (FL) emerges as an effective way to enable AI over massively distributed nodes without uploading the raw data. Conventional works mostly focus on learning a single unified model for one solitary task. Multitask learning (MTL) outperforms single-task learning by training multiple models concurrently, leading to reduced model sizes and increased flexibility. However, existing FL efforts often face challenges in efficiently managing MTL scenarios, particularly with the presence of stragglers, without incurring prohibitive computation and communication costs.In this article, inspired by the natural cloud-base station (BS)-terminal hierarchy of cellular networks, we provide a viable resource-aware hierarchical federated MTL (RHFedMTL) solution to meet the task heterogeneity corresponding to different nonindependent and identically distributed (IID) training data sets. Specifically, a primal-dual method has been leveraged to effectively transform the coupled MTL into some local optimization subproblems within BSs. Therefore, it enables solving different tasks within a BS and aggregating the multitask result in the cloud without uploading the raw data. Furthermore, compared with existing methods that reduce resource costs by simply changing the aggregation frequency, we dive into the intricate relationship between resource consumption and learning accuracy, and develop a resource-aware learning strategy for adjusting the iteration number on local terminals and BSs to meet the resource budget. Extensive simulation results demonstrate the effectiveness and superiority of RHFedMTL in terms of improving the learning accuracy and boosting the convergence rate.**

***Index Terms*—Artificial intelligence (AI), federated learning (FL), mobile edge computing, multitask learning (MTL).**

Manuscript received 7 February 2024; revised 18 March 2024; accepted 16 April 2024. Date of publication 23 April 2024; date of current version 9 July 2024. This work was supported in part by the Zhejiang Key Research and Development Plan under Grant 2022C01093; in part by the National Natural Science Foundation of China under Grant 62071425; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR23F010005; in part by the National Key Laboratory of Wireless Communications Foundation under Grant 2023KP01601; and in part by the Big Data and Intelligent Computing Key Laboratory of CQUPT under Grant BDIC-2023-B-001. This article was presented in part at the IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Kyoto, Japan, 2022, [DOI: 10.1109/PIMRC54779.2022.9977670]. *(Corresponding author: Rongpeng Li.)*

Xingfu Yi and Rongpeng Li are with the College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: yixingfu@zju.edu.cn; lirongpeng@zju.edu.cn).

Chenghui Peng, Fei Wang, and Jianjun Wu are with the Wireless Communications Lab, Huawei Technologies Company Ltd., Shanghai 200122, China (e-mail: pengchenghui@huawei.com; wangfei76@huawei.com; wujianjun@huawei.com).

Zhifeng Zhao is with Zhejiang Lab, Hangzhou 311121, China (e-mail: zhaozf@zhejianglab.com).

Digital Object Identifier 10.1109/JIOT.2024.3392584

I. Introduction

BENEFITING from the rapid development of cellular networks, artificial intelligence (AI) over massive Internet of Things (IoT) or smartphones becomes possible, and there is a growing consensus that 6G will be revolutionary as a network of AI [2], [3], [4], [5]. Meanwhile, network AI in 6G should simultaneously allow various sensing, mining, prediction, and reasoning tasks across different industries. For example, autonomous driving algorithms are assembled with a series of coupled tasks, including detection, tracking, and mapping in perception; and motion and occupancy forecast in prediction [6]. Therefore, there emerges a common concern for storing and transmitting rapidly expanded data through the network in the near future [7]. Besides, the heterogeneous and complex nature of Industrial IoT (IIoT) presents many technical challenges, such as privacy, heterogeneity, and resource efficiency [8].

In that regard, federated learning (FL) only requires uploading the training gradients instead of the clients' raw data [9] and has been extensively used in many scenarios [10], [11], [12]. The concept of FL has garnered significant attention. Exemplified by the well-known federated averaging (FedAvg) algorithm [9], traditional FL approaches, however, face challenges in scalability and efficiency, particularly when dealing with large-scale, heterogeneous networks [13]. Therefore, there emerges a substantial body of works to improve FL. For example, by generalizing and reparametrizing FedAvg [9], FedProx [14] is introduced to tackle heterogeneity in FL. Liu et al. [15] devised a client-edge-cloud hierarchical FL (HFL) system to swiftly strike a balance between computational efficiency, model accuracy, and data privacy. Yang et al. [16] proposed an energy-efficient computation and transmission resource allocation scheme over wireless networks. However, most industry solutions deploy standalone models for tasks tailored to different geographical areas [6]. These FL approaches [9], [10], [11], [12], [13], [14], [15], [16] primarily focus on training a single unified model, and lack the essential efficiency to learn heterogeneous tasks from different nonindependent and identically distributed (IID) data sets across multiple terminals.

As for federated multitask learning (FedMTL), Smith et al. [17] extended the FL framework to support simultaneous multitask learning (MTL), and introduced a MOCHA algorithm with enhanced personalization and boosted performance. However, in MOCHA [17], one task

corresponds to one terminal, and it turns computation-intensive and even impractical. Marfoq et al. [18] investigated federated MTL under the assumption that each data distribution can be regarded as a mixture of several unknown underlying distributions, and proposed the expectation-maximization (EM)-based FedEM algorithm on top of stochastic gradient descent (SGD) for a hierarchical learner–client–server structure. Nevertheless, due to its excessively certain assumption for underlying distributions, FedEM still fails to meet the required scalability for large-scale IoT scenarios.

Therefore, inspired by the natural hierarchy of cellular networks, which spans from cloud, base stations (BSs) to terminals,[1] we propose a resource-aware hierarchical FedMTL (RHFedMTL) method, so as to meet the task heterogeneity corresponding to different non-IID training data sets. In particular, each BS is linked with several terminals and responsible for training a specific learning task with the support of attached terminals, while the cloud oversees the aggregation of multitask results. Moreover, by employing a primal-dual optimization method, we transform the global primal optimization problem into separate dual subproblems distributed across the BSs. Besides, RHFedMTL uses the convergence-verified stochastic dual coordinate ascent (SDCA) [19], [20]. This division of the problem also lays a crucial foundation for developing a resource-aware learning strategy, evidenced by the established relationship between learning accuracy and the number of iterations, as well as the tradeoff between computations on terminals and BSs. In summary, compared with the existing works, our main contributions are summarized as follows.

1) We leverage a hierarchical structure for FedMTL on top of the primal-dual-based SDCA, and propose an RHFedMTL method, which boosts the flexibility of MTL and enables coupled MTL over massive terminals without uploading their raw data.
2) We provide the derived convergence bound of the vanilla hierarchical FedMTL (HFedMTL), and unveil the relationship between the terminal iteration number and the BS iteration number required to converge.
3) On top of the aforementioned derived relationship, we propose a resource-aware algorithm RHFedMTL, which dynamically adapts the terminal iteration number (and corresponding BS iteration number) to balance the tradeoff between learning accuracy and resource cost, to maintain the learning performance with limited resource budget.
4) Through extensive simulations, we evaluate the superior performance of the RHFedMTL algorithm in terms of its effectiveness, robustness, and resource-awareness.

The remainder of this article is organized as follows. In Section II, we discuss the related works. In Section III, we introduce the details of the system model for FedMTL and present the formulated resource-constrained problem. In Section IV, we derive the dual problem for the vanilla HFedMTL algorithm. Based on the convergence analysis, we give the RHFedMTL algorithm and discuss how to dynamically set the terminal iteration number based on the provided convergence analysis. Section V talks about the simulation scenarios and demonstrates the numerical results. We conclude this article in Section VI.

[1]Notably, in this article, considering the diversified terminologies in the literature like [18], a "terminal" is not fully equivalent to a "client."

II. Related Works

As a distributed machine learning framework, FL is privacy-friendly since it only requires uploading the training gradients instead of the clients' raw data [9], [21]. Traced back to FedAvg by McMahan et al. [9], FL has since witnessed diverse applications in areas demanding ameliorated data privacy, such as healthcare and finance [11], [22]. Meanwhile, there are more works in optimizing the resource overhead of conventional FL. For example, Yang et al. [16] proposed an energy-efficient computation and transmission resource allocation scheme in wireless networks. Chen et al. [23] proposed a probabilistic user selection scheme to reduce the FL convergence time and the FL training loss. Dinh et al. [24] proposed a resource allocation algorithm over wireless networks to capture the tradeoff between the wall clock training time and the terminal energy consumption. Regarding resource utilization optimization in FL for wireless networks, Yang et al. [25] primarily developed an analytical model to characterize FL performance. Liu and Simeone [26] studied the adaptive power allocation for distributed gradient descent. However, these works primarily consider single-task learning.

On the other hand, most industry solutions deploy standalone models for each task [6]. For vehicle networking scenarios where autonomous vehicles can simultaneously train on different perception & prediction tasks and tailored to different geographical areas, single-task strategies inevitably add to the need for intensive computations and data volume for effective processing and transmission. Instead of training a single unified model across different terminals, FedMTL extends FL to simultaneously learn multiple tasks, enabling more personalized and efficient training. The inception of FedMTL can be attributed to the broader field of MTL [27], which demonstrated the effectiveness of leveraging shared representations across coupled tasks to improve model performance and generalization in fields like natural language processing and computer vision [28], [29]. Smith et al. [17] showed that MTL is naturally suited to handle the statistical challenges in training machine learning models over distributed networks of devices and proposed a novel systems-aware optimization method, MOCHA, that is robust to practical systems issues. Marfoq et al. [18] introduced FedEM, designed for both client-server frameworks and fully decentralized environments, on the premise that each local data set can be modeled as a combination of several unknown underlying distributions. In scenarios where there are L such underlying distributions, FedEM's objective is to derive L unified component parameters across the relevant clients, with each client maintaining and refining its own set of L unique component parameters and their corresponding mixture weights.

Recent developments in FedMTL have also focused on improving resource efficiency and scalability. For example,

TABLE I
NOTATIONS

Notations	Description
N	Number of BSs
B_b	Index of One BS
N_b	Number of Terminals under One BS
$T_{b,t}$	Index of Terminal under One BS
n_b	Number of Data under One BS
$n_{b,t}$	Number of Data in Terminal t under BS b
$\boldsymbol{w}_b$	Model Parameters of Task b
$\boldsymbol{x}_b^i$	One Train Data under Task b
$\boldsymbol{y}_b^i$	One Train Label under Task b
$\boldsymbol{\alpha}_b$	Dual Parameters of All Data under Task b
$\boldsymbol{\alpha}_{[t]}$	Dual Parameters of Local Data in Terminal t
$\mathcal{R}_b$	MTL Regulation Function
$\boldsymbol{r}_b$	MTL Regulation Parameters
$\mathbf{W}$	Model Parameters of All Tasks
$\mathbf{R}$	MTL Regulation Functions of All Tas
Θ_b	The Lower Bound of Duality Gap Update under BS b
Θ	The Lower Bound of Duality Gap Update
ϵ_D	Convergence Target of Duality Gap
J	Type of Considered Resources
C_{bud}	Resource Budget
C_{tol}	Total Resource Consumption
C_{dev}	Standard Resource Cost of Single Terminal
C_{real}	Resource Cost of Single Terminal
M	No. of Multi-task Server Aggregation
K	No. of BS Iterations
H_b	No. of Terminal Iterations under BS b
$C_{j,\text{bud}}$	Resource Budget
$C_{j,\text{dev}}$	Cost for Terminal Iteration
$C_{j,\text{BS}}$	Cost for BS Iteration
γ	The Smoothness of Loss Function
λ_1	Weight of Self Regulation
λ_2	Weight of Multi-task Regulation

Caldas et al. [30] attempted to address the challenge of terminal resource constraints in FedMTL, and proposed methods like lossy compression and federated dropout to reduce the computational and communication burden on terminals, making FedMTL more accessible and scalable. Furthermore, Wei et al. [31] explored the integration of differential privacy with FedMTL to enhance the privacy-preserving aspects of FedMTL. However, these efforts neglect the hierarchical network structure and cannot cope with the heterogeneity challenge [13]. On the other hand, in order to address the heterogeneity challenge [13], HFL is conjectured upon the standard FL framework, by adopting a layered approach for model training. Typically, HFL involves local model training on edge devices, regional aggregators, and a unified model updating globally [15]. Besides, Liu et al. [15] demonstrated the efficacy of HFL in handling heterogeneous data across different layers and improving learning outcomes in large-scale networks. However, a limitation in HFL studies belongs to the possibly unrealistic assumption of the existence of only a single task within the network. Therefore, originally designed for single-task applications, these frameworks show constrained effectiveness in MTL environments [9], [14], [15], given the exacerbated challenge in cases with stragglers that lag in processing or communication due to MTL's heightened demands.

In conclusion, efforts toward HFL and FedMTL are disentangled, while factors, such as the network hierarchy and communication efficiency, are not given sufficient consideration until recent works in [17] and [18]. Notably, the proposed RHFedMTL possesses significant differences from MOCHA [17] and FedEM [18]. In RHFedMTL, each BS connects to multiple terminals and oversees the training of a distinct learning task, aided by the connected terminals, with the cloud coordinating the aggregation of results from multiple tasks. This setting significantly contrasts with MOCHA [17], wherein one task corresponds to one terminal. Moreover, despite the partial resemblance to the hierarchical learner–client–server setting in FedEM [18], RHFedMTL uses SDCA [19], [20], [32] rather than SGD in FedEM, since the former offers stronger convergence results than primal-only methods (e.g., SGD) for the same iteration cost [19], [20]. Meanwhile, RHFedMTL does not assume the existence of a distinct mixture of distributions for the local data set. Following our previous work on the HFedMTL algorithm that allows massive nodes from distributed areas to join in the federated multi-task learning process [1], RHFedMTL incorporates a resource-aware hierarchical resource management scheme, which takes account of the limitation of the resource budget and capably adjusts the learning process of local terminals to balance the tradeoff between resource cost and MTL performance. To our best knowledge, this belongs to the very first resource-aware, HFedMTL approach and makes a significant difference with existing literature in [15], [33], [34], and [35].

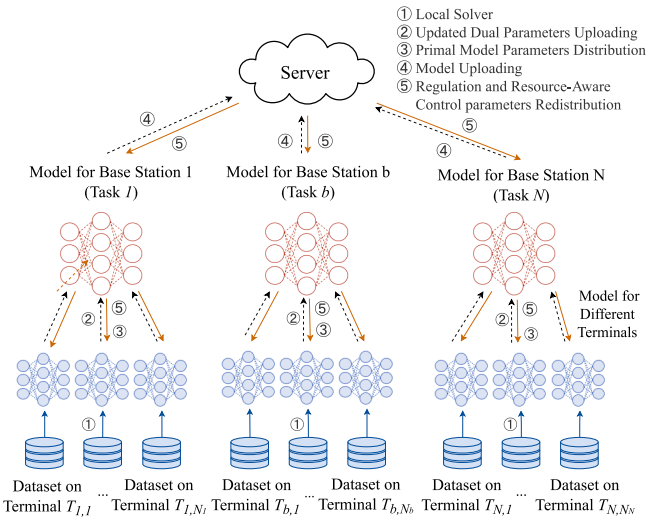


Fig. 1. System model of RHFedMTL.

III. System Model and Problem Formulation

A. System Model

Beforehand, some key notations in this paper are provided in Table I. Inspired by [17] and [18], we primarily consider a resource-intense hierarchical mobile computing environment with multiple task models to be learned. In particular, as shown in Fig. 1, we assume there exist N BSs (i.e., $B_1, \ldots, B_N$) connected to the cloud server, each corresponding to one of the N coupled tasks. Meanwhile, each BS B_b, $b \in \{1, \ldots, N\}$ covers N_b terminals $T_{b,t}$, $t \in \{1, \ldots, N_b\}$ (e.g., smartphones, IoT devices), with each terminal collecting $S_{b,t}$ samples of

data. The MTL aims to learn the model parameters $\boldsymbol{w}_b$, for each of the N tasks, using local data sets, that is

$$\min_{\mathbf{W},\mathbf{R}} \left\{ \frac{1}{N} \sum_{b=1}^{N} \left(\frac{1}{n_b} \sum_{i=1}^{n_b} \mathcal{L}\left(\boldsymbol{w}_b^{\intercal} \boldsymbol{x}_b^i\right) + \frac{\lambda_1}{2} \|\boldsymbol{w}_b\|^2 + \mathcal{R}_b(\boldsymbol{w}_b) \right) \right\}$$
$$\text{s.t. } y_b^i = \boldsymbol{w}_b^{\intercal} \boldsymbol{x}_b^i \quad \forall i \in \{1, \ldots, n_b\}, b \in \{1, \ldots, N\} \tag{1}$$

where $n_b \triangleq \sum_{t=1}^{N_b} S_{b,t}$ represents the total number of data distributed under BS b (i.e., task b) and the superscript $\intercal$ indicates the transpose operator. $y_b^i \triangleq \boldsymbol{w}_b^{\intercal} \boldsymbol{x}_b^i$ corresponds to the output for input of $\boldsymbol{x}_b^i$ in local data set under BS B_b. In summary, $\mathbf{W} \triangleq [\boldsymbol{w}_1 \cdots \boldsymbol{w}_{N_b}]$ and $\mathbf{R} \triangleq [\mathcal{R}_1, \ldots, \mathcal{R}_b, \ldots, \mathcal{R}_N]$ represent the weights of all BSs and the MTL regulation function for all tasks, respectively.

Consistent with the single-task learning, the MTL in (1) imposes an $\mathcal{L}_2$ regulation with constant $\lambda_1 > 0$. Nevertheless, MTL adds regulation functions $\mathcal{R}_b(\boldsymbol{w}_b)$ to reflect the relationship among tasks. Typically, assumptions on MTL regulation functions can be categorized into two groups, dependent on the a-priori existence of the relationships amongst tasks [17]. Specifically, [36], [37] assume that the coupling structure between different tasks is known a priori while [38] assumes unknown yet learnable relationships between different tasks. As for the former case, following [39], we could directly adopt an $\mathcal{L}_2$ regulation on $\mathbf{W}$ to constrain differences among models for different tasks while reflecting their potential resemblance (to a reference task). Mathematically, such a regulation could be formulated as

$$\mathcal{R}_b(\boldsymbol{w}_b) = \frac{\lambda_2}{2} \|\boldsymbol{w}_b - \boldsymbol{r}\|^2 \tag{2}$$

where $\boldsymbol{r} \triangleq (1/N) \sum_{b=1}^{N} \boldsymbol{w_b}$ and λ_2 indicates the relative importance of the multitask regulation to the overall loss in (1). Meanwhile, as for the latter case, the multitask regulation function $\mathcal{R}_b$ can be set as

$$\mathcal{R}_b(\boldsymbol{w}_b) = \frac{1}{N} \lambda_2 \text{tr}\left(\boldsymbol{W} \boldsymbol{\Omega}^{-1} \boldsymbol{W}^{\intercal} \right) \tag{3}$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix. Furthermore, the multitask relationship matrix $\boldsymbol{\Omega}$ in (3) can be learned from data. In this work, we primarily focus on the former case in (2) while our work can be extended to the latter one with an additional learning loop.

B. Problem Formulation

In a resource-intense mobile computing environment, it is of vital importance to limit the amount of resources to achieve a target error of the loss function, so as to keep the operational cost low on the basis of no system backlog. Hence, the formulation in (1) shall be complemented with the resource constraints. Consistent with [35], the terminology of "resources" here is generic and includes time, energy & economic costs related to both computation and communication. Without loss of generality, assume that there exists J types of resources and the resource cost for the same type of terminals is equal. $C_{j,\text{tol}}$ and $C_{j,\text{bud}}$, for $j \in \{1, \ldots, J\}$, represent the total resource consumption and the budget of type-j resource, respectively. We mainly focus on the resource consumption of terminals and BSs due to their natural importance in the mobile network. Besides, the standard resource consumption of type-j resource for a terminal and a BS to perform an iteration step is defined as $C_{j,\text{dev}}$ and $C_{j,\text{BS}}$, respectively. Furthermore, assume that a hierarchical iteration methodology is adopted here. In other words, one server iteration includes multiple BS iterations, while one BS iteration encompasses several terminal iterations. Consequently, on the basis of M server iterations, for K BS iterations and H_b terminal iterations under certain BS B_b ($b \in \{1, \ldots, N\}$), the type-$j \in \{1, \ldots, J\}$ resource consumption shall be bounded as

$$KM \sum_{b=1}^{N} \left(C_{j,\text{BS}} + N_b H_b C_{j,\text{dev}} \right) \leq C_{j,\text{bud}}. \tag{4}$$

In other words, the MTL problem in (1) can be reformulated as (5). In this article, in order to solve this coupled MTL, we attempt to develop a resource-aware federated solution that adheres to the privacy-friendly federated policy (i.e., no raw data uploading) and involves a learning methodology with appropriate parameters K and H_b.

IV. Resource-Aware Hierarchical Federated Multitask Learning

A. Dual Formulation

The coupled MTL models make it challenging to directly compute the parameters without knowing all the data sets distributed among terminals. In this part, inspired by the primal-dual methodology, we present the approach to formulate the dual formulation of (5) and decompose the global problem into individual localized subproblems to be independently solved by terminals.

We define $\boldsymbol{\alpha}$ as the concatenation of all the dual variables α_b^i. Mathematically, $\boldsymbol{\alpha} \triangleq [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_b, \ldots, \boldsymbol{\alpha}_N]$, where $\boldsymbol{\alpha}_b$ is the concatenation of all the dual variables under the same BS and can be represented in two interchangeable forms: 1) $\boldsymbol{\alpha}_b \triangleq [\alpha_b^1, \ldots, \alpha_b^{n_b}]$ is a simple combination of all the dual variables distributed among various terminals under one BS,

$$\min_{\mathbf{W},\mathbf{R}} \left\{ \frac{1}{N} \sum_{b=1}^{N} \left(\frac{1}{n_b} \sum_{i=1}^{n_b} \mathcal{L}\left(\boldsymbol{w}_b^{\intercal} \boldsymbol{x}_b^i\right) + \frac{\lambda_1}{2} \|\boldsymbol{w}_b\|^2 + \mathcal{R}_b(\boldsymbol{w}_b) \right) \right\} \quad \forall K, H_b \in \{1, \ldots, \infty\}$$
$$\text{s.t.} \quad KM \sum_{b=1}^{N} \left(C_{j,\text{BS}} + N_b H_b C_{j,\text{dev}} \right) \leq C_{j,\text{bud}} \quad \forall j \in \{1, \ldots, J\} \ \forall b \in \{1, \ldots, N\} \tag{5}$$
$$y_b^i = \boldsymbol{w}_b^{\intercal} \boldsymbol{x}_b^i \quad \forall i \in \{1, \ldots, n_b\}, b \in \{1, \ldots, N\}$$

where α_b^i is the dual variable for the data point $(\boldsymbol{x}_b^i, y_b^i)$ and 2) $\boldsymbol{\alpha}_b \triangleq [\boldsymbol{\alpha}_{[1]}, \ldots, \boldsymbol{\alpha}_{[t]}, \ldots, \boldsymbol{\alpha}_{[N_b]}]$ where $\boldsymbol{\alpha}_{[t]} \in \mathbb{R}^{S_{b,t}}$ is the concatenation of dual variables corresponding to the local data set of terminal $T_{b,t}$.

We employ the following lemma to demonstrate the equivalence between the primal problem and the aggregation of decomposed subproblems at each BS, thus facilitating the subsequent analysis of a HFedMTL solution.

Lemma 1: For any $\boldsymbol{\alpha}$, we have

$$D(\boldsymbol{\alpha}) = \frac{1}{N}\sum_{b=1}^{N}\left(\frac{1}{n_b}\sum_{i=1}^{n_b} -\mathcal{L}^*\left(-\alpha_b^i\right) + \mathcal{R}_b^*(\boldsymbol{w}_b)\right) = \frac{1}{N}\sum_{b=1}^{N} D_b(\boldsymbol{\alpha}_b) \tag{6}$$

where

$$D_b(\boldsymbol{\alpha}_b) \triangleq \frac{1}{n_b}\sum_{i=1}^{n_b} -\mathcal{L}^*\left(-\alpha_b^i\right) + \frac{\lambda_1\lambda_2\|\boldsymbol{r}_b\|^2 - \|\boldsymbol{A}_b\boldsymbol{\alpha}_b\|^2 - 2\lambda_2\boldsymbol{A}_b\boldsymbol{\alpha}_b\boldsymbol{r}_b^{\mathsf{T}}}{2(\lambda_1+\lambda_2)} \tag{7}$$

and $\boldsymbol{A}_b \in \mathbb{R}^{d\times n_b}$ collects data examples $A_i = 1/n_b\boldsymbol{x}_b^i$ in its columns. Given dual variables $\boldsymbol{\alpha}_b$, corresponding primal variables can be found via $\boldsymbol{w}_b = [1/(\lambda_1+\lambda_2)](\lambda_2\boldsymbol{r}_b + \boldsymbol{A}_b\boldsymbol{\alpha}_b)$. Moreover, $\mathcal{L}^*$ and $\mathcal{R}_b^*$ are the conjugate dual functions of $\mathcal{L}$ and $\mathcal{R}_b$, respectively.

Proof: The proof is a direct application of the definition of the conjugate function and we leave it in Appendix. ■

Lemma 1 suggests that the MTL problem can be decomposed and solved as several subproblems by individual BSs. Next, we resort to tackling each subproblem of (7) separately. In that regard, as *coordinate ascent* methods require no step size and have a well-defined stopping criterion given by the duality gap [19], we use SDCA which has proven very suitable for use in large-scale problems, and give stronger convergence results than the primal-only methods (e.g., SGD) at the same iteration cost [32]. Given the formulation of (7), terminals can find updates $\Delta\alpha_b^i$ to the dual variables in $\boldsymbol{\alpha}_b$ by accessing only the locally stored data $(\boldsymbol{x}_b^i, y_b^i)$. In other words, the terminals under the same BS can solve the related part of the subproblem locally and independently, and the algorithm could converge at a higher speed compared to gradient descent methods. Therefore, with selected data $(\boldsymbol{x}_{b,t}^i, y_{b,t}^i)$ in the training process, by Lemma 1, terminal-oriented subproblem can be formulated as

$$\max_{\Delta\alpha_b^i}\left\{-\mathcal{L}^*\left(-(\alpha_b^i+\Delta\alpha_b^i)\right) + \frac{\lambda_2}{2}\|\boldsymbol{r}_b\|^2 - \frac{\lambda_1+\lambda_2}{2}\left[\boldsymbol{w}_b + \frac{1}{n_b(\lambda_1+\lambda_2)}\sum_{i=1}^{S_{b,t}}\Delta\alpha_b^i\boldsymbol{x}_{b,t}^i\right]^2\right\}. \tag{8}$$

Finally, the MTL problem in (6) has been split into several subproblems across distributed BSs, each of which can be tackled by terminals locally. Correspondingly, a HFedMTL solution, which involves a hierarchical iteration mechanism, including the server iteration, the BS iteration, and the terminal iteration, can be attained.

1) As for each terminal iteration of the training process, terminals conduct their local training process to maximize the local optimizing function (8) and communicate with their BS periodically.
2) During each BS iteration, the BSs update and distribute model parameters by aggregating intermediate variables uploaded by terminals and then upload their model to the server at a lower frequency.
3) Based on the received models, the server updates the parameters of the regulation function $\mathcal{R}_b$ and sends the updated parameters back to terminals through their connected BSs. During the server iteration, the server generates and sends the regulation parameters to all terminals (via BSs) using the uploaded model parameters of all tasks.

In summary, for each server iteration, the BSs perform K BS iterations, and during each iteration of BS B_b (where $b \in 1, \ldots, N$), the terminals conduct H_b terminal iterations.

B. Convergence Analysis

We first present the convergence analysis of HFedMTL, which will guide the design of the following resource-aware solution, before delving into the details of resource-aware HFedMTL. Beforehand, we give a definition of the duality gap after one terminal iteration.

Definition 1: Within each BS iteration, we define the duality gap of subproblems specifying how far we are from the optimum on terminal t with all other terminals fixed. Mathematically, given $\boldsymbol{\alpha}_{[1]} \ldots \boldsymbol{\alpha}_{[N_b]}$ from N_b terminals

$$\mathcal{E}_{D_{b,t}}(\boldsymbol{\alpha}_b) \triangleq \max_{\hat{\boldsymbol{\alpha}}_{[t]}} D_b\left(\boldsymbol{\alpha}_{[1]}, \ldots, \hat{\boldsymbol{\alpha}}_{[t]}, \ldots, \boldsymbol{\alpha}_{[N_b]}\right) - D_b\left(\boldsymbol{\alpha}_{[1]}, \ldots, \boldsymbol{\alpha}_{[t]}, \ldots, \boldsymbol{\alpha}_{[N_b]}\right) \tag{9}$$

We also have the following assumption of the updated duality gap.

Assumption 1: We assume that there exists $\Theta_b \in (0, 1]$ such that for any given $\boldsymbol{\alpha}_b$, the subproblem running on terminal t alone returns a (possibly random) update $\Delta\boldsymbol{\alpha}_{[t]}$ after H_b terminal iterations such that the expectation of updated duality gap is bounded as

$$\mathbf{E}\left[\mathcal{E}_{D_{b,t}}\left(\boldsymbol{\alpha}_{[1]}, \ldots, \boldsymbol{\alpha}_{[t]} + \Delta\boldsymbol{\alpha}_{[t]}, \ldots, \boldsymbol{\alpha}_{[N_b]}\right)\right] \le \Theta_b \cdot \mathcal{E}_{D_{b,t}}(\boldsymbol{\alpha}_b). \tag{10}$$

Assume that the loss functions $\mathcal{L}$ is $(1/\gamma)$-smooth. Then, consistent with [19, Proposition 1]

$$\Theta_b = \left(1 - \frac{(\lambda_1+\lambda_2)n_b\gamma}{1+(\lambda_1+\lambda_2)n_b\gamma}\frac{1}{\tilde{n}_b}\right)^{H_b} \tag{11}$$

where $\tilde{n}_b \triangleq \max_t n_{b,t}$ is the size of the largest local data set among terminals. Specifically, $\Theta_b \triangleq 1$ means that the terminals under task b made no updates and therefore no resource consumption, while $\Theta_b \to 0$ implies that the duality gap comes to zero, which is unrealistic as the number of terminal iterations (and correspondingly the resource consumption) grows to infinity in this case. Therefore, Θ_b indicates the resource required by the system. We also define $\Theta \triangleq \max_b \Theta_b$ to facilitate subsequent operations.

The following theorem gives the convergence analysis of HFedMTL with respect to the number of BS iterations K.

Theorem 1: Assume the loss functions $\mathcal{L}$ is $(1/\gamma)$-smooth, choosing BS iteration number K such that

$$K > \left(1-(1-\Theta)\frac{\eta^*}{T^*}\right)\log\frac{\sum_{b=1}^{N} n_b}{N\epsilon_D} \tag{12}$$

we have

$$\mathbf{E}\Big[D(\boldsymbol{\alpha}^*) - D\Big(\boldsymbol{\alpha}^{(K)}\Big)\Big] \le \epsilon_D \tag{13}$$

where ϵ_D is the target convergence duality gap, $T^* \triangleq \max_b N_b$, $\eta^* \triangleq \min_b \eta_b$, $\eta_b = [(\lambda_1+\lambda_2)\gamma/n_b\sigma + (\lambda_1+\lambda_2)\gamma]$ with $\sigma \ge \max_b \sigma_{b\min}$ and

$$\sigma_{b\min} \triangleq \max_{\alpha} n_b^2 \frac{\sum_{t=1}^{N_b}\left\|\boldsymbol{A}_{[t]}\boldsymbol{\alpha}_{[t]}\right\|^2 - \|\boldsymbol{A}_b\boldsymbol{\alpha}_b\|^2}{\|\boldsymbol{\alpha}_b\|^2}. \tag{14}$$

Proof: According to Lemma 1, the value of dual problem after $(k+1)$th BS iteration can be expressed as

$$D\Big(\boldsymbol{\alpha}^{(k+1)}\Big) = \frac{1}{N}\sum_{b=1}^{N} D_b\left(\boldsymbol{\alpha}_b^{(k)} + \frac{1}{N_b}\sum_{t=1}^{N_b}\Delta\boldsymbol{\alpha}_{[t]}\right). \tag{15}$$

Recalling that D is the pointwise infimum of a family of affine functions of $\boldsymbol{\alpha}$ as shown in Appendix, D is concave [40]. Using the concavity of dual function $D(\boldsymbol{\alpha})$

$$D\Big(\boldsymbol{\alpha}^{(k+1)}\Big) \ge \frac{1}{N}\sum_{b=1}^{N}\frac{1}{N_b}\sum_{t=1}^{N_b} D_b\Big(\boldsymbol{\alpha}_b^{(k)} + \Delta\boldsymbol{\alpha}_{[t]}\Big). \tag{16}$$

Denoting $\hat{\alpha}_{[t]}^*$ to be the local maximizer as in (9), we have

$$\begin{aligned}
&\mathbf{E}\Big[D\Big(\boldsymbol{\alpha}^{(k+1)}\Big) - D\Big(\boldsymbol{\alpha}^{(k)}\Big)\Big] \\
&\ge \frac{1}{N}\sum_{b=1}^{N}\frac{1}{N_b}\sum_{t=1}^{N_b}\Big\{D_b\Big(\boldsymbol{\alpha}_b^{(k)} + \Delta\boldsymbol{\alpha}_{[t]}\Big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big)\Big\} \\
&\ge \frac{1}{NN_b}\sum_{b=1}^{N}\sum_{t=1}^{N_b}\Big\{D_b\Big(\boldsymbol{\alpha}_b^{(k)} + \Delta\boldsymbol{\alpha}_{[t]}\Big) \\
&\quad - D_b\Big(\boldsymbol{\alpha}_{[1]}^{(k)},\ldots,\hat{\boldsymbol{\alpha}}_{[t]}^*,\ldots,\boldsymbol{\alpha}_{[N_b]}^{(k)}\Big) \\
&\quad + D_b\Big(\boldsymbol{\alpha}_{[1]}^{(k)},\ldots,\hat{\boldsymbol{\alpha}}_{[t]}^*,\ldots,\boldsymbol{\alpha}_{[N_b]}^{(k)}\Big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big)\Big\} \\
&\ge \frac{1}{NN_b}\sum_{b=1}^{N}\sum_{t=1}^{N_b}\Big\{\mathcal{E}_{D_{b,t}}\Big(\boldsymbol{\alpha}_b^{(k)}\Big) - \mathcal{E}_{D_{b,t}}\Big(\boldsymbol{\alpha}_b^{(k)} + \Delta\boldsymbol{\alpha}_{[t]}\Big)\Big\}.
\end{aligned} \tag{17}$$

Meanwhile, under Assumption 1

$$\begin{aligned}
&\mathbf{E}\Big[D\Big(\boldsymbol{\alpha}^{(k+1)}\Big) - D\Big(\boldsymbol{\alpha}^{(k)}\Big)\Big|\boldsymbol{\alpha}^{(k)}\Big] \\
&\ge \frac{(1-\Theta)}{N}\sum_{b=1}^{N}\frac{1}{N_b}\sum_{t=1}^{N_b}\mathcal{E}_{D_{b,t}}\Big(\boldsymbol{\alpha}_b^{(k)}\Big).
\end{aligned} \tag{18}$$

Recalling the definition of $D_b(\boldsymbol{\alpha}_b)$ in (7), we can obtain (19) shown at the bottom of the page, where the equalities (a) to (c) come from Lemma 1. Besides, the inequality (d) is due to (20) shown at the bottom of the next page, in [19, Th. 2] with σ in (14), and the inequality (e) is given by introducing an extra $\eta_b \in [0,1]$ to link $\boldsymbol{\alpha}_b^*$ (i.e., the maximizer of (19)). After

$$\begin{aligned}
&\sum_{t=1}^{N_b}\mathcal{E}_{D_{b,t}}\Big(\boldsymbol{\alpha}_b^{(k)}\Big) \\
&\overset{(a)}{=} \max_{\hat{\boldsymbol{\alpha}}_b}\left\{\sum_{t=1}^{N_b}\Big[D_b\Big(\boldsymbol{\alpha}_{[1]}^{(k)},\ldots,\hat{\boldsymbol{\alpha}}_{[t]},\ldots,\boldsymbol{\alpha}_{[N_b]}^{(k)}\Big) - D_b\Big(\boldsymbol{\alpha}_{[1]}^{(k)},\ldots,\boldsymbol{\alpha}_{[t]}^{(k)},\ldots,\boldsymbol{\alpha}_{[N_b]}^{(k)}\Big)\Big]\right\} \\
&\overset{(b)}{=} \max_{\hat{\boldsymbol{\alpha}}_b}\left\{\frac{1}{n_b}\sum_{i=1}^{n_b}\Big(-\mathcal{L}^*\big(-\hat{\alpha}_b^i\big) + \mathcal{L}^*\Big(-\alpha_b^{i^{(k)}}\Big)\Big) + \frac{1}{2(\lambda_1+\lambda_2)}\sum_{t=1}^{N_b}\Big(-2\lambda_2\boldsymbol{A}_{[t]}\Big(\hat{\boldsymbol{\alpha}}_{[t]} - \boldsymbol{\alpha}_{[t]}^{(k)}\Big)\boldsymbol{r}_b^{\mathsf{T}}\right. \\
&\qquad \left. - \Big\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)} + \boldsymbol{A}_{[t]}\Big(\hat{\boldsymbol{\alpha}}_{[t]} - \boldsymbol{\alpha}_{[t]}^{(k)}\Big)\Big\|^2 + \Big\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}\Big\|^2\Big)\right\} \\
&\overset{(c)}{=} \max_{\hat{\boldsymbol{\alpha}}_b}\left\{D_b\big(\hat{\boldsymbol{\alpha}}_b\big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big) + \frac{\big\|\boldsymbol{A}_b\hat{\boldsymbol{\alpha}}_b\big\|^2 - \|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}\|^2}{2(\lambda_1+\lambda_2)}\right. \\
&\qquad \left. + \frac{1}{2(\lambda_1+\lambda_2)}\sum_{t=1}^{N_b}\Big[-\Big\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)} + \boldsymbol{A}_{[t]}\Big(\hat{\boldsymbol{\alpha}}_{[t]} - \boldsymbol{\alpha}_{[t]}^{(k)}\Big)\Big\|^2 + \Big\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}\Big\|^2\Big]\right\} \qquad (19) \\
&\overset{(d)}{\ge} \max_{\hat{\boldsymbol{\alpha}}_b}\left\{D_b\big(\hat{\boldsymbol{\alpha}}_b\big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big) - \frac{\sigma}{2(\lambda_1+\lambda_2)n_b^2}\Big\|\hat{\boldsymbol{\alpha}}_b - \boldsymbol{\alpha}_b^{(k)}\Big\|^2\right\} \\
&\overset{(e)}{\ge} \max_{\eta_b\in[0,1]}\left\{D_b\Big(\eta_b\boldsymbol{\alpha}_b^* + (1-\eta_b)\boldsymbol{\alpha}_b^{(k)}\Big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big) - \frac{\sigma}{2(\lambda_1+\lambda_2)n_b^2}\Big\|\eta_b\boldsymbol{\alpha}_b^* + (1-\eta_b)\boldsymbol{\alpha}_b^{(k)} - \boldsymbol{\alpha}_b^{(k)}\Big\|^2\right\} \\
&\overset{(f)}{\ge} \max_{\eta_b\in[0,1]}\left\{\eta_b D_b\big(\boldsymbol{\alpha}_b^*\big) + (1-\eta_b)D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big) + \frac{\gamma\eta_b(1-\eta_b)}{2n_b}\Big\|\boldsymbol{\alpha}_b^* - \boldsymbol{\alpha}_b^{(k)}\Big\|^2 - \frac{\eta_b^2\sigma}{2(\lambda_1+\lambda_2)n_b^2}\Big\|\boldsymbol{\alpha}_b^* - \boldsymbol{\alpha}_b^{(k)}\Big\|^2\right\} \\
&\overset{(g)}{\ge} \max_{\eta_b\in[0,1]}\left\{\eta_b\Big(D_b\big(\boldsymbol{\alpha}_b^*\big) - D_b\Big(\boldsymbol{\alpha}_b^{(k)}\Big)\Big) + \frac{\eta_b}{2n_b}\left(\gamma(1-\eta_b) - \frac{\eta_b\sigma}{(\lambda_1+\lambda_2)n_b}\right)\Big\|\boldsymbol{\alpha}_b^* - \boldsymbol{\alpha}_b^{(k)}\Big\|^2\right\}
\end{aligned}$$

applying the property of $(1/\gamma)$-smooth function and simple mathematical manipulations, we have the inequalities (f) and (g), respectively.

Letting $(\gamma(1-\eta_b)-[\eta_b\sigma/(\lambda_1+\lambda_2)n_b])=0$, we have $\eta_b=[(\lambda_1+\lambda_2)n_b\gamma/n_b\sigma+(\lambda_1+\lambda_2)n_b\gamma]$. Thus

$$\sum_{t=1}^{N_b}\mathcal{E}_{D_{b,t}}\left(\boldsymbol{\alpha}_b^{(k)}\right)\geq\eta_b\left(D_b\left(\boldsymbol{\alpha}_b^*\right)-D_b\left(\boldsymbol{\alpha}_b^{(k)}\right)\right). \tag{21}$$

Substituting (21) into (18), we can derive

$$\begin{aligned}&\mathbf{E}\left[D\left(\boldsymbol{\alpha}^{(k+1)}\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)|\boldsymbol{\alpha}^{(k)}\right]\\&\geq\frac{(1-\Theta)}{N}\sum_{b=1}^{N}\frac{1}{N_b}\eta_b\left(D_b\left(\boldsymbol{\alpha}^*\right)-D_b\left(\boldsymbol{\alpha}^{(k)}\right)\right)\\&\geq\beta\left(D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)\right)\end{aligned} \tag{22}$$

where $\beta\triangleq(1-\Theta)\eta^*/T^*$, $T^*\triangleq\max_b N_b$ and $\eta^*\triangleq\min_b\eta_b$. Therefore

$$\begin{aligned}&\mathbf{E}\left[D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(k+1)}\right)\right]\\&=\mathbf{E}\left[D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)-\left(D\left(\boldsymbol{\alpha}^{(k+1)}\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)\right)\right]\\&\leq(1-\beta)\left(D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)\right).\end{aligned} \tag{23}$$

Thus, we have

$$\mathbf{E}\left[D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(k)}\right)\right]\leq(1-\beta)^k\left(D\left(\boldsymbol{\alpha}^*\right)-D\left(\boldsymbol{\alpha}^{(0)}\right)\right).$$

Consistent with [17], we use the bound on the initial duality gap proved in [20, Lemma 10], which states that $D_b(\boldsymbol{\alpha}_b^*)-D_b(\boldsymbol{\alpha}_b^{(0)})\leq n_b$. Therefore

$$\epsilon_D\leq(1-\beta)^K\frac{1}{N}\sum_{b=1}^{N}n_b. \tag{24}$$

Finally, since when $x>0$, $x>\log x$, we have

$$K\geq\frac{\log\frac{N\epsilon_D}{\sum_{b=1}^{N}n_b}}{\log(1-\beta)}=\frac{\log\frac{\sum_{b=1}^{N}n_b}{N\epsilon_D}}{\log\frac{1}{1-\beta}}>(1-\beta)\log\frac{\sum_{b=1}^{N}n_b}{N\epsilon_D}.$$

■

Remark: Theorem 1 shows that the value of β is mostly affected by the hardest task and the learning problem will always converge after an update of $\mathcal{R}_b$ if we set the number for BS iterations K sufficiently large. Hence, HFedMTL is robust to the change of MTL parameters and guaranteed to converge.

Together with (11), Theorem 1 implies the following corollary.

Corollary 1: The minimal K increases given a decrease of terminal iteration number H_b, so there exists a tradeoff between the terminal iteration number H_b and BS iteration number K.

C. Resource-Aware Implementation

To meet all the requirements of the resource-aware problem defined in (5), we extend the vanilla HFedMTL method to a resource-aware approach, so as to optimize the system model at a minimum resource cost.

Unlike HFedMTL, which predefines all parameters, the resource-aware HFedMTL method is aware of the costs and budget of the system resources, so as to dynamically adjust the terminal iteration number H_b under BS B_b, $b\in\{1,\ldots,N\}$ and BS iteration number K while meeting the convergence target of duality gap in (13). Recall Theorem 1, which states that a larger terminal iteration number H_b linearly increases resource consumption, but decreases the incurred duality gap nonlinearly. Given the resource budget, there must exist a range of feasible H_b, within which choosing a bigger H_b will increase the terminal iteration cost, but the decreased BS iteration number K required for convergence will decrease the overall resource consumption. The following theorem verifies the aforementioned intuitions.

Theorem 2: Assume that the loss function $\mathcal{L}$ is $(1/\gamma)$-smooth, for any convergence target ϵ_D, in order to solve the MTL problem in (5), if there exists a terminal iteration number H_b, $b\in\{1,\ldots,N\}$ satisfying (25), shown at the bottom of the page, the problem is feasible.

Proof: Considering the constraint in (5), a direct analysis leads to that the iteration number K under BS B_b should satisfy

$$K\leq\left[\min_{j\in\{1,\ldots,J\}}C_{j,\text{bud}}\left[M\sum_{b=1}^{N}\left(C_{j,\text{BS}}+N_bH_bC_{j,\text{dev}}\right)\right]^{-1}\right]. \tag{26}$$

Taking account the convergence analysis in Theorem 1

$$K>\left(1-(1-\Theta)\frac{\eta^*}{T^*}\right)\log\frac{\sum_{b=1}^{N}n_b}{N\epsilon_D}. \tag{27}$$

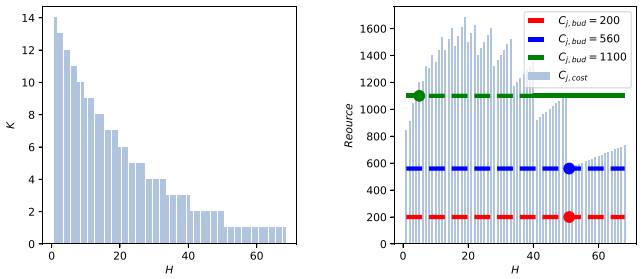
Rearranging (26) and (27), we can derive that

$$\begin{aligned}&\left(1-(1-\Theta)\frac{\eta^*}{T^*}\right)\log\frac{\sum_{b=1}^{N}n_b}{N\epsilon_D}\\&<\left[\min_{j\in\{1,\ldots,J\}}C_{j,\text{bud}}\left[M\sum_{b=1}^{N}\left(C_{j,\text{BS}}+N_bH_bC_{j,\text{dev}}\right)\right]^{-1}\right].\end{aligned} \tag{28}$$

Substituting (11) into (28), we have the theorem. ■

$$\max_{\hat{\boldsymbol{\alpha}}_b}\left\{\left\|\boldsymbol{A}_b\hat{\boldsymbol{\alpha}}_b\right\|^2-\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}\|^2+\sum_{t=1}^{N_b}\left[-\left\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}+\boldsymbol{A}_{[t]}\left(\hat{\boldsymbol{\alpha}}_{[t]}-\boldsymbol{\alpha}_{[t]}^{(k)}\right)\right\|^2+\left\|\boldsymbol{A}_b\boldsymbol{\alpha}_b^{(k)}\right\|^2\right]\right\}\geq\max_{\hat{\boldsymbol{\alpha}}_b}\left\{-\frac{\sigma}{n_b^2}\left\|\hat{\boldsymbol{\alpha}}_b-\boldsymbol{\alpha}_b^{(k)}\right\|^2\right\} \tag{20}$$

$$M\sum_{b=1}^{N}\left(C_{j,\text{BS}}+N_bH_bC_{j,\text{dev}}\right)\left\{1-\frac{\eta^*}{T^*}\left[1-\left(1-\frac{(\lambda_1+\lambda_2)n_b\gamma}{1+(\lambda_1+\lambda_2)n_b\gamma}\frac{1}{\tilde{n}_b}\right)^{\min_b H_b}\right]\right\}\log\frac{\sum_{b=1}^{N}n_b}{N\epsilon_D}<C_{j,\text{bud}} \tag{25}$$



(a) (b)

Fig. 2. BS iteration number K and resource cost $C_{j,\text{cost}}$ until convergence target ϵ_D over different settings of $H_b = H$, $b \in \{1, \ldots, N\}$. (a) Required K for convergence. (b) $C_{j,\text{cost}}$ until the convergence target.

Remark: Theorem 2 provides a feasible range of H_b, which will help determine an appropriate terminal iteration number. On the basis of that, the chosen BS iteration number will potentially reduce the overall resource consumption.

As given in Theorem 2, the left side of (25) represents the needed resource cost to meet the convergence target ϵ_D (13) under a certain number of terminal iterations. In case where $N = 5$, $N_b = 5$, and $\epsilon_D = 0.01$, we demonstrate in Fig. 2 the numerical relationship between the BS iteration number and the resource cost with respect to a unanimous terminal iteration number $H_b = H$, $b \in \{1, \ldots, N\}$. Accordingly, with limited resource budget $C_{j,\text{bud}}$, there exist three possible cases and they are represented by horizontal lines of different colors in Fig. 2, wherein solid lines mean $C_{j,\text{bud}} > C_{j,\text{cost}}$ while dashed lines mean $C_{j,\text{bud}} < C_{j,\text{cost}}$. Besides, the dot indicates the value of the chosen H.

1) *Case 1:* The given resource budget $C_{j,\text{bud}}$ is too limited to converge, just as shown in the red line of Fig. 2. In this case, RHFedMTL capably traverses all the range of H_b to minimize the resource cost of terminals (i.e., to be most resource efficient) and reduce the number of BS iterations to meet the limited resource budget.
2) *Case 2:* The given resource budget $C_{j,\text{bud}}$ is just enough to converge, just as shown in the blue line of Fig. 2. In this case, RHFedMTL capably traverses all the feasible range of H_b to satisfy (25), making a balance between learning speed and efficiency. Correspondingly, it sets the BS iteration number according to (12).
3) *Case 3:* The given resource budget $C_{j,\text{bud}}$ leads to a resource surplus, just as shown in the green line of Fig. 2. In this case, RHFedMTL can choose an appropriately smaller valid number of H_b, so as to fully leverage the resource budget. Notably, the adaption of H_b also implies the ability to cope with the straggler issue.

Finally, under the RHFedMTL framework in Algorithm 1, we present the LocalDualMethod in Procedure 1.

Algorithm 1: Resource-Aware HFedMTL Method

Data: $(\boldsymbol{x}^i_{b,t}, y^i_{b,t})$, $t \in (1, 2, \ldots, N_b)$, $b \in (1, \ldots, N)$. Each terminal $T_{b,t}$ contains a local data set containing $S_{b,t}$ samples of data
Input: $C_{j,\text{dev}}$, $C_{j,\text{BS}}$, $C_{j,\text{bud}}$
Initialize : $\boldsymbol{\alpha}_b^{(0)} \triangleq 0$, $\boldsymbol{w}_b \triangleq 0$, $\boldsymbol{r}_b \triangleq 0$ $\forall b \in (1, \ldots, N)$

for $j = 0, 1, \ldots$ **do** server iteration
  server send regulation parameters $\boldsymbol{r}_b$ to terminals through their connected BSs after gathering the uploaded information
  for *BS (i.e., tasks)* B_b, $b \in (0, 1, \ldots, N)$ *in parallel* **do**
    send T^*, η^* to all terminals
    for $k = 0, 1, \ldots, K$ **do** BS iteration
      for *all terminals* $T_{b,t}$, $t \in \{1 \cdots N_b\}$ *in parallel* **do**
        $(\Delta\boldsymbol{\alpha}_{[t]}, \Delta\boldsymbol{w}_{b,t}) \leftarrow$ ResourceSavingMethod $(\boldsymbol{\alpha}_{[t]}, \boldsymbol{w}_b, \boldsymbol{r}_b, C_{j,\text{dev}}, C_{j,\text{BS}}, C_{j,\text{bud}}, T^*, \eta^*)$
      $\boldsymbol{w}_b \leftarrow \boldsymbol{w}_b + \frac{1}{N_b}(\sum_{t=1}^{N_b} \Delta\boldsymbol{w}_{b,t})$
  update regulation parameters $\boldsymbol{r}_b \leftarrow \frac{1}{N}\sum_{b=1}^{N} \boldsymbol{w_b}$
  update $T^* \leftarrow \max_b N_b$, $\eta^* \leftarrow \min_b \frac{(\lambda_1+\lambda_2)\gamma}{n_b\sigma+(\lambda_1+\lambda_2)\gamma}$.
Output: $\boldsymbol{w}_b$, $\boldsymbol{r}_b$ $\forall b \in (1, \ldots, N)$

Procedure 1: LocalDualMethod

Data: Local data $\{(\boldsymbol{x}^i_{b,t}, y^i_{b,t})\}_{i=1}^{S_{b,t}}$
Input: $\boldsymbol{\alpha}_{[t]}$, $\boldsymbol{w}_b$, $\boldsymbol{r}_b$, $C_{j,\text{dev}}$, $C_{j,\text{BS}}$, $C_{j,\text{bud}}$, T^*, η^*
Initialize : $\Delta\boldsymbol{\alpha}_{[t]} \leftarrow 0$, $\Delta\boldsymbol{w}_{b,t} \leftarrow 0$, $\boldsymbol{w}_{b,t} \leftarrow \boldsymbol{w}_b$, $f(H) \triangleq (N_b H C_{j,\text{dev}} + N C_{j,\text{BS}})(1 - \frac{\eta^*}{T^*}[1 - (1 - \frac{(\lambda_1+\lambda_2)n_b\gamma}{1+(\lambda_1+\lambda_2)n_b\gamma}\frac{1}{\tilde{n}_b})^H]) \log\frac{\sum_{b=1}^{N} n_b}{N\epsilon_D}$

$H \leftarrow 0$, $H^{\text{cad}} \leftarrow 0$, $c \leftarrow +\infty$
for $h = S_{b,t}, \ldots, 1$ **do**
  if $f(h) < c$ **then**
    $c \leftarrow f(h)$
    $H^{\text{cad}} \leftarrow h$
  if $f(h) \le C_{j,bud}$ ***and*** $f(h+1) > C_{j,bud}$ **then**
    $H \leftarrow h$
if $H = 0$ **then**
  $H \leftarrow H^{\text{cad}}$
for $h = 0, \ldots, H$ **do** terminal iteration
  choose i uniformly at random in local dataset
  find $\Delta\alpha_b^i$ of $\boldsymbol{\alpha}_b$ to maximize the local optimizing function (8)
  $\Delta\boldsymbol{\alpha}_{[t]} \leftarrow (\Delta\boldsymbol{\alpha}_{[t]})^i + \Delta\alpha_b^i$
  $\boldsymbol{w}_{b,t} \leftarrow \boldsymbol{w}_{b,t} + \frac{1}{\lambda_1+\lambda_2}\Delta\alpha_b^i \boldsymbol{x}^i_{b,t}$
Output: $\Delta\boldsymbol{\alpha}_{[t]}$, $\Delta\boldsymbol{w}_{b,t} \triangleq \frac{1}{\lambda_1+\lambda_2}(\frac{1}{S_{b,t}}\sum_{i=1}^{S_{b,t}} x^i_{b,t}\alpha^i_b)$

V. Simulation and Numerical Results

In this part, we illustrate the performance of the proposed RHFedMTL algorithm. Consistent with the methodology in MOCHA [17], we analyze accelerometer and gyroscope data from 30 participants engaged in six distinct activities, including walking variations, sitting, standing, and lying down [41]. Utilizing 561-feature vectors that capture both time and frequency domain information for each sample, we treat each participant as a unique task. The objective is to differentiate between sitting and other activities based on these comprehensive feature vectors. Meanwhile, the training data set of one task is distributed among terminals while the test data set is stored in BS for performance evaluation.

In order to express the overall performance of the system more intuitively, we calculate the overall accuracy by averaging the accuracy of all separate tasks on their test data sets. For simplicity, we primarily consider $J = 1$ type of resources and consider energy as the single resource type in our experiments. The resource cost includes the computational

TABLE II
DEFAULT SETTINGS

Notations	Default Setting
H	2
$C_{j,\text{bud}}$	$1,400$
$C_{j,\text{dev}}$	0.1
$C_{j,\text{BS}}$	10
N	5
N_b	5
γ	1
λ_1	10^{-4}
λ_2	10^{-6}

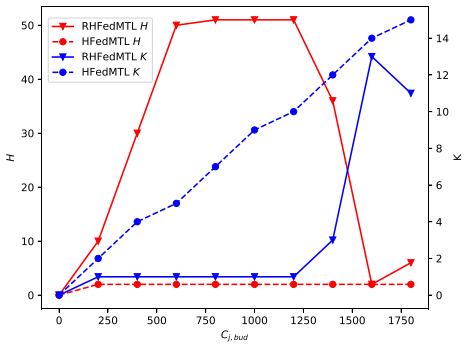


Fig. 3. Chosen H and K under different resource budget $C_{j,\text{bud}}$.

TABLE III
MODEL ACCURACY WITH DIFFERENT RESOURCE BUDGETS FOR DIFFERENT ALGORITHMS

$C_{j,\text{bud}}$	RHFedMTL	HFedMTL	FedAvg	FedProx	FedEM
200	0.70	0.68	0.54	0.53	0.54
400	0.73	0.62	0.58	0.52	0.67
600	0.72	0.63	0.54	0.55	0.66
800	0.72	0.71	0.61	0.55	0.66
$1,000$	0.73	0.65	0.59	0.58	0.66
$1,200$	0.77	0.68	0.56	0.57	0.66
$1,400$	0.78	0.74	0.55	0.65	0.65
$1,600$	0.74	0.70	0.55	0.62	0.65

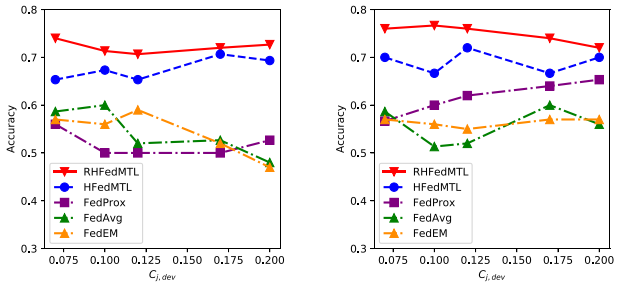


Fig. 4. Model accuracy over terminal iteration cost $C_{j,\text{dev}}$. (a) $C_{j,\text{bud}} = 400$. (b) $C_{j,\text{bud}} = 1400$.

cost for terminal iteration (i.e., local training process) and BS iteration (i.e., aggregation and transmission), consistent with (5). We compare RHFedMTL with FedMTL methods, such as vanilla HFedMTL [1] and FedEM [18], as well as federated single-task learning methods like FedAvg [9] and FedProx [14]. In particular, we simulate federated single-task learning by learning averaged BS models independently. Besides, in the context of FedEM [18], each client manages L learners, with every learner designed to adapt to different underlying distributions' parameters. Therefore, the client and learner in FedEM are somewhat equivalent to BS and terminal in our work. Hence, the task quantity L in FedEM corresponds to the terminal number N_b (i.e., $L = N_b$), while the task quantity in RHFedMTL is linked to the number of BSs N. In other words, FedEM can only be fairly compared by setting $N = N_b = L$. Accordingly, we incorporate FedEM into the performance comparison only when $N = N_b = 5$, as illustrated in Figs. 4 and 5, while omitting it for cases $N \neq N_b$, showcased in Figs. 6 and 7. The system parameters are summarized in Table II.

Our initial experiment investigates how the terminal iteration number H_b correlates with the BS iteration number K, and validates the consistency with (13). For simplicity of representation, we set $H \triangleq \min_b H_b$ as shown in (25) and set H_b to be the same as H for all BSs. The experiment results are shown in Fig. 2. In line with our previous discussions [see (11) and (25)], there exists a negative correlation between the terminal iteration number H and the BS iteration number K. However, the relationship between H and the overall system resource cost does not consistently follow a monotonic pattern. Hence, when the cost associated with the BS, denoted as $C_{j,\text{BS}}$, is relatively high, decreasing the BS iteration number K may offset the resource expenses incurred by an increase in the terminal iteration number H. The numerical outcomes for selected values of H and K, tailored to various resource budgets $C_{j,\text{dev}}$, are depicted in Fig. 3. Owing to its limitation in resource adaptability, HFedMTL employs a constant terminal iteration number and compensates by escalating the BS iteration number to fully utilize the resource budget. Conversely, as lately substantiated by the findings in Fig. 6 and Table III, RHFedMTL demonstrates the capability to fine-tune both H and K concurrently, achieving superior outcomes within a specified resource allocation.

Fig. 4 explores the impact of terminal iteration costs, $C_{j,\text{dev}}$, on model accuracy. As expected, under particular resource constraints, the high cost associated with $C_{j,\text{dev}}$ leads to a reduction in terminal iterations, thus decreasing the learning accuracy. In scenarios of limited resources, FedEM outperforms FedAvg and FedProx, demonstrating the benefits of MTL. Besides, with more sufficient resources, federated single-task learning methods exhibit more significant performance gains. However, regardless of varying resource levels, RHFedMTL consistently delivers superior results, affirming its robust performance.

In Fig. 5, we present the accuracy for individual tasks, where the resource budget $C_{j,\text{bud}} = 1,400$ and the number of tasks, $N = 5$. Besides, the model accuracy under various resource budget conditions is detailed in Table III. It can be observed from Fig. 5 and Table III that compared to single-task learning methods like FedProx and FedAvg, which falter on certain tasks, MTL techniques, such as RHFedMTL and HFedMTL, capitalize on the coupling among tasks and secure a more consistent overall performance. Besides, although FedEM demonstrates stable and quick convergence,

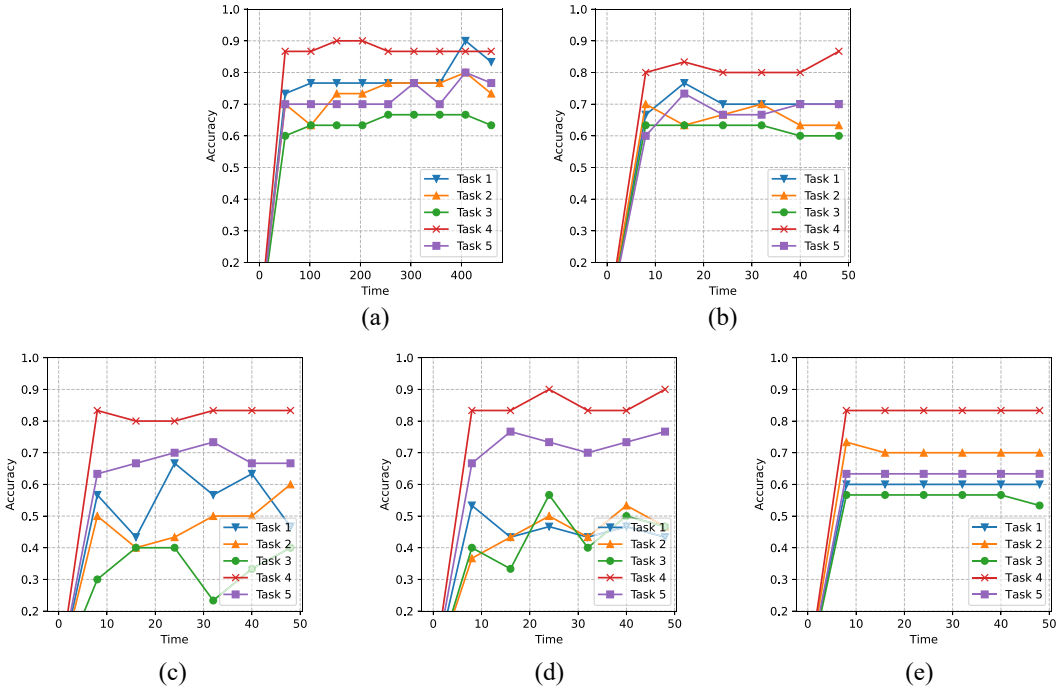


Fig. 5. Task accuracy with $N = 5$, $C_{j,\text{bud}} = 1400$. (a) RHFedMTL. (b) HFedMTL. (c) FedAvg. (d) FedProx. (e) FedEM.

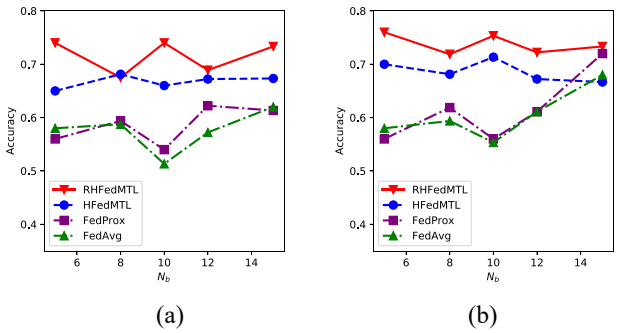


Fig. 6. Model accuracy versus different terminal number N_b. (a) $C_{j,\text{bud}} = 400$. (b) $C_{j,\text{bud}} = 1400$.

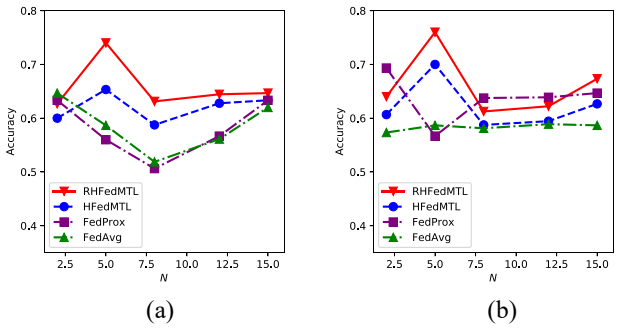


Fig. 7. Model accuracy for different task number N. (a) $C_{j,\text{bud}} = 400$. (b) $C_{j,\text{bud}} = 1400$.

its proficiency in exploiting task interrelations falls short of RHFedMTL. Moreover, consistent with previous discussions, RHFedMTL outperforms HFedMTL in terms of learning accuracy.

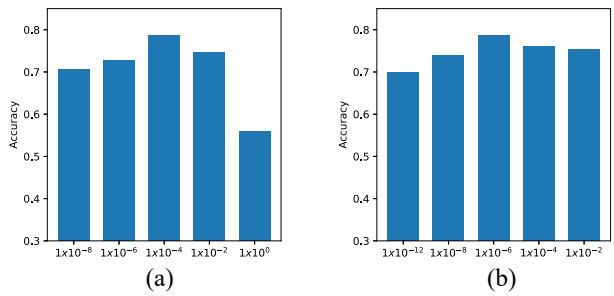


Fig. 8. Model accuracy of RHFedMTL with different settings of regulation parameters. (a) λ_1. (b) λ_2.

Furthermore, we vary N or N_b to assess the impact of the number of terminals or BSs on model accuracy. Notably, given the finite data set size, an increase in N or N_b results in a reduction of data allocated to each terminal. In Fig. 6, we vary the number of terminals linked to a BS, ranging from 5 to 15. Consistent with our previous discussions, MTL approaches outperform single-task learning methods. However, along with the abundance of system resources, the efficiency of single-task methodologies, especially FedProx, gets better. In Fig. 7, we adjust the number of BSs (and equivalently the task number) within a range of 2 to 15. Notably, under constrained resources (e.g., $C_{j,\text{bud}} = 400$), FedProx and FedAvg exhibit comparable results. With a moderate increase in resources, FedProx demonstrates consistent performance enhancements over FedAvg. In contrast, RHFedMTL, equipped with a resource-aware mechanism, generally achieves superior outcomes.

Subsequently, we delve into the sensitivity of RHFedMTL's model accuracy to variations in the regularization parameters λ_1 and λ_2. The findings, depicted in Fig. 8, reveal that

the model's accuracy diminishes when the self-regulation parameter λ_1 is set too low, compromising its self-regulatory function. Conversely, an excessively high λ_1 can also impair performance. A similar pattern is observed with the multitask regulation parameter λ_2. Consequently, we select default values as $\lambda_1 = 10^{-4}$ and $\lambda_2 = 10^{-6}$, respectively, to effectively balance performance and regulation.

VI. Conclusion

In this article, we have addressed the importance of network AI and proposed an RHFedMTL framework based on the primal-dual method for tackling federated MTL problems with stragglers. In particular, RHFedMTL has considered the hierarchy of cellular networks and encompassed a three-tier iteration mechanism, including server iteration, BS iteration, and terminal iteration. Moreover, the primal-dual method SDCA has been leveraged to effectively transform the coupled MTL into some local optimization subproblems within BSs. We have analyzed the convergence bound of the proposed framework, and derived a guiding relationship between terminal iteration and BS iteration. Afterwards, we have developed a resource-aware learning strategy for local terminals and BSs to obtain more satisfactory learning performance under a given resource budget. Extensive experimentation results have demonstrated the effectiveness and robustness of the proposed method.

Appendix
Proof of Lemma 1

Proof: Recalling the relationship $y_b^i \triangleq \boldsymbol{w}_b^\mathsf{T}\boldsymbol{x}_b^i$ in (1), the Lagrangian dual of the problem could be derived as follows:

$$
\begin{aligned}
&D(\boldsymbol{\alpha}) \\
&= \inf_{\mathbf{W}, y_b^i} \frac{1}{N}\sum_{b=1}^{N}\Bigg\{\frac{\lambda_1}{2}\|\boldsymbol{w}_b\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}_b - \boldsymbol{r}_b\|^2 \\
&\qquad + \frac{1}{n_b}\sum_{i=1}^{n_b}\left(\mathcal{L}\left(y_b^i\right) + \alpha_b^i\left(y_b^i - \boldsymbol{w}_b^\mathsf{T}\boldsymbol{x}_b^i\right)\right)\Bigg\} \\
&= \inf_{y_{b,t}^i} \frac{1}{N}\sum_{b=1}^{N}\frac{1}{n_b}\sum_{i=1}^{n_b}\left(\mathcal{L}\left(y_b^i\right) + \alpha_b^i y_b^i\right) + \inf_{\mathbf{W}} \frac{1}{N}\sum_{b=1}^{N} \\
&\quad \left(\frac{\lambda_1}{2}\|\boldsymbol{w}_b\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}_b - \boldsymbol{r}_b\|^2 - \frac{1}{n_b}\sum_{i=1}^{n_b}\alpha_b^i \boldsymbol{w}_b^\mathsf{T}\boldsymbol{x}_b^i\right) \\
&= -\frac{1}{N}\sum_{b=1}^{N}\frac{1}{n_b}\sum_{i=1}^{n_b}\sup_{y_b^i}\left(-\alpha_b^i y_{b,t}^i - \mathcal{L}\left(y_b^i\right)\right) + \inf_{\mathbf{W}} \frac{1}{N}\sum_{b=1}^{N} \\
&\quad \left(\frac{\lambda_1}{2}\|\boldsymbol{w}_b\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}_b - \boldsymbol{r}_b\|^2 - \frac{1}{n_b}\sum_{i=1}^{n_b}\alpha_b^i \boldsymbol{w}_b^\mathsf{T}\boldsymbol{x}_b^i\right) \\
&= \frac{1}{N}\sum_{b=1}^{N}\frac{1}{n_b}\sum_{i=1}^{n_b} -\mathcal{L}^*\left(-\alpha_b^i\right) \\
&\quad + \inf_{\mathbf{W}} \frac{1}{N}\sum_{b=1}^{N}\left(\frac{\lambda_1}{2}\|\boldsymbol{w}_b\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}_b - \boldsymbol{r}_b\|^2 - \frac{1}{n_b}\sum_{i=1}^{n_b}\alpha_b^i \boldsymbol{w}_b^\mathsf{T}\boldsymbol{x}_b^i\right).
\end{aligned} \tag{29}
$$

Since the optimization of $\boldsymbol{w}_b$ in (29) is independent of the models in other tasks, after taking the gradient of $\boldsymbol{w}_b$ in the second term and setting it to zero, for any model $\boldsymbol{w}_b$ we would have

$$
\boldsymbol{w}_b = \frac{1}{\lambda_1 + \lambda_2}\left(\lambda_2 \boldsymbol{r}_b + \frac{1}{n_b}\sum_{i=1}^{n_b}\alpha_b^i \boldsymbol{x}_b^i\right). \tag{30}
$$

Substituting the chosen $\boldsymbol{w}_b$ in (30) into (29), we get the lemma. ■

References

[1] X. Yi, R. Li, C. Peng, J. Wu, and Z. Zhao, "HFedMTL: Hierarchical federated multi-task learning," in *Proc. IEEE Int. Symp. Personal, Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2022, pp. 1–6.

[2] L. Zhang, Y.-C. Liang, and D. Niyato, "6G visions: Mobile ultra-broadband, super Internet of Things, and artificial intelligence," *China Commun.*, vol. 16, no. 8, pp. 1–14, Aug. 2019.

[3] R. Li et al., "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[4] R. Li, Z. Zhao, X. Xu, F. Ni, and H. Zhang, "The collective advantage for advancing communications and intelligence," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 96–102, Aug. 2020.

[5] R. Li, W. Liang, C. Peng, X. An, Z. Zhao, and H. Zhang, "Network AI management & orchestration: A federated multi-task learning case," in *Proc. IEEE Global Commun. Conf. (GLOBECOM) Workshops*, Madrid, Spain, Dec. 2021, pp. 1–6.

[6] Y. Hu et al., "Planning-oriented autonomous driving," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, BC, Canada, Jun. 2023, pp. 1–10.

[7] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[8] W. Z. Khan, M. Rehman, H. M. Zangoti, M.K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Elect. Eng.*, vol. 81, Jan. 2020, Art. no. 106522.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, Apr. 2017, pp. 1–10.

[10] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[12] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, 2019, pp. 1387–1395.

[13] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," 2023, *arXiv:2301.01299*.

[14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[15] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

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

[17] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Los Angeles, CA, USA, Dec. 2017, pp. 1–19.

[18] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 15434–15447.

[19] M. Jaggi et al., "Communication-efficient distributed dual coordinate ascent," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, Dec. 2014, pp. 1–15.

[20] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *J. Mach. Learn. Res.*, vol. 18, pp. 1–49, Jul. 2018.

[21] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.
[22] M. J. Sheller et al., "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 12598.
[23] 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, Apr. 2021.
[24] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Feb. 2021.
[25] 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.
[26] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 170–185, Jan. 2021.
[27] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, Jul. 1997.
[28] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*.
[29] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, 2021.
[30] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," 2018, *arXiv:1812.07210*.
[31] K. Wei et al., "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[32] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res.*, vol. 14, no. 2, pp. 567–599, Feb. 2013.
[33] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.
[34] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Nat. Acad. Sci.*, vol. 118, no. 17, 2021, Art. no. e2024789118.
[35] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[36] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Seattle, WA, USA, Aug. 2004, pp. 109–117.
[37] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2006, pp. 1–8.
[38] L. Jacob, J.-P. Vert, and F. Bach, "Clustered multi-task learning: A convex formulation," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 21, 2008, pp. 1–14.
[39] Y. Zhou, R. Jin, and S. C.-H. Hoi, "Exclusive lasso for multi-task feature selection," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, Sardinia, Italy, May 2010, pp. 988–995.
[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2004.
[41] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21th Int. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, Bruges, Belgium, Apr. 2013.

Xingfu Yi received the B.E. degree in communication engineering from Tongji University, Shanghai, China, in 2021, and the M.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2024.

Rongpeng Li (Senior Member, IEEE) received the B.E. degree in communication engineering from Xidian University, Xi'an, China, in 2010, and the Ph.D. degree in communication and information systems from Zhejiang University, Hangzhou, China, in 2015.

He is currently an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was a Research Engineer with the Wireless Communication Laboratory, Huawei Technologies Company, Ltd., Shanghai, China, from August 2015 to September 2016. He was a Visiting Scholar with the Department of Computer Science and Technology, University of Cambridge, Cambridge, U.K., from February 2020 to August 2020. His research interest currently focuses on networked intelligence for communications evolving.

Dr. Li received the Wu Wenjun Artificial Intelligence Excellent Youth Award. He serves as an Editor for *China Communications*.

Chenghui Peng received the B.E. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2001.

He is an Expert Researcher of Wireless Technology Lab, Huawei Technologies Co. Ltd., Shenzhen, China. His main research directions include 6G wireless network architecture, task-oriented native intelligent architecture, mobile computing network, and wireless distributed learning paradigm. He has applied for more than 100 patents in the LTE, and 5G and 6G field.

Fei Wang received the Doctoral degree from the University of Science and Technology of China, Hefei, China, in 2015.

He is currently the Chief Researcher of Wireless Technology Lab, Huawei Technologies Co. Ltd., Shenzhen, China. His main research directions include 6G wireless network architecture, wireless distributed learning paradigm, federated learning, and large model.

Jianjun Wu received the M.S. degree from Southwest Jiaotong University, Chengdu, China, in 2001.

He was the Chief Researcher and the Director of the Future Network Laboratory, Huawei Technologies Co. Ltd., Shenzhen, China, when the manuscript was initially submitted. His main research direction was future wireless network architecture, include 6G network architecture definition, 5G E2E slicing solution research, standards, and industry development. He was the Director of the European Research Center Branch of Huawei 2012 Laboratories and led the local team to fully participate in the definition and research of 5G origins, such as 5GIA and 5GPPP. He initiated and successfully established the 5GAA and 5GACIA industry alliances.

Zhifeng Zhao (Member, IEEE) received the B.E. degree in computer science, the M.E. degree in communication and information systems, and the Ph.D. degree in communication and information systems from PLA University of Science and Technology, Nanjing, China, in 1996, 1999, and 2002, respectively.

From 2002 to 2004, he acted as a Postdoctoral Researcher with Zhejiang University, Hangzhou, China, where his researches were focused on multimedia next-generation networks and soft switch technology for energy efficiency. From 2005 to 2006, he acted as a Senior Researcher with PLA University of Science and Technology, where he performed research and development on advanced energy-efficient wireless routers, ad-hoc network simulators, and cognitive mesh networking test bed. From 2006 to 2019, he was an Associate Professor with the College of Information Science and Electronic Engineering, Zhejiang University. He is currently the Chief Engineering Officer with Zhejiang Lab, Hangzhou. His research areas include software-defined networks, mobile and wireless networks, computing networks, and collective intelligence.

Dr. Zhao is the Symposium Co-Chair of ChinaCom 2009 and 2010. He is the Technical Program Committee Co-Chair of the 10th IEEE International Symposium on Communication and Information Technology (ISCIT 2010).