Semi-Federated Learning Accelerated by Over-the-Air Distortion

Jingheng Zheng, Hui Tian, *Senior Member, IEEE,* Wanli Ni, *Member, IEEE,*
Yang Tian, and Ping Zhang, *Fellow, IEEE*

***Abstract*—In this paper, we study a semi-federatd learning (SemiFL) framework that uses the computational resources of both the base station and devices to perform hybrid federated and split learning, wherein over-the-air computation is employed to enable communication-efficient gradient aggregation. Unlike previous works that focus on mitigating aggregation distortion during wireless transmission, this research explores the utilization of over-the-air distortion to accelerate the convergence of SemiFL. Specifically, the acceleration effect is achieved by intentionally permitting a high mean square error (MSE) threshold in the non-stable region of SemiFL, so as to amplify the ratio of the power scaling factor to the normalizing factor. Then, we derive a lower bound for the expected loss reduction between two consecutive rounds in the non-stable region, which reveals that high distortion helps accelerate convergence by boosting loss descent. However, the upper bound of the optimality gap indicates that suppressing over-the-air distortion is beneficial to improve the final convergence in the stable region. Given the antagonistic roles of over-the-air distortion in different regions, we propose a two-region MSE threshold configuration scheme by setting a high MSE threshold in the non-stable region and reducing it in the stable region. Based on this scheme, two distinct energy consumption minimization problems are formulated. Accordingly, we propose two resource allocation algorithms with closed-form solutions. Simulation results show that over-the-air distortion can significantly accelerate the convergence of SemiFL compared to state-of-the-art benchmarks. Moreover, energy consumption can be reduced by using the proposed optimization algorithms.**

***Index Terms*—Federated learning, over-the-air computation, distortion utilization, convergence acceleration, resource allocation.**

I. Introduction

Past few years have witnessed the thriving emergence of artificial intelligence (AI)-enabled applications, such as unmanned vehicles, smart healthcare, and AI-empowered Internet of Things [1], [2]. When deploying these advanced applications in the sixth-generation (6G) wireless network, a critical challenge is how to efficiently train high-quality AI models [3]. The conventional centralized learning (CL) framework necessitates collecting all available data from devices to train a global model [4]. Nonetheless, this centralized approach raises privacy leakage issues owing to the transmission of massive raw data. Due to its ability to preserve privacy, federated learning (FL) has garnered substantial research interests [5]. As a distributed framework, FL deployed in wireless networks allows devices collaboratively train a shared global model by aggregating local gradients at the base station (BS), thereby preserving data privacy. Recently, the data collection and storage capabilities of local devices have been significantly enhanced [6]. Substantial available data impose overwhelming local training burdens, whereas the BS's strong computational capabilities remain underutilized in conventional FL schemes. Hence, it is necessary to design a new model training framework that utilizes resources rationally to balance the workload between the BS and devices.

A. Motivations

By synthesizing FL and CL into an implement-efficient framework, the existing semi-federated learning (SemiFL) enables the BS to capitalize on its powerful computational capability to execute CL concurrently with FL across devices [7]–[10]. Although SemiFL alleviates workload on local devices, partially uploading local datasets still poses privacy leakage risks. In the literature, techniques such as mixup [11] and fully homomorphic encryption [12] can mitigate privacy leakage. However, the mixup technique simply superposes local data using a set of normalized weight coefficients, offering privacy protection to only a limited extent. As for fully homomorphic encryption, it requires considerable local computation due to the complicated encryption and decryption operations, limiting its implementations on resource-constrained devices. In view of the drawbacks of exiting methods, another key question is that: *How can we ameliorate the SemiFL framework to ensure efficient and privacy-preserving data uploading*?

Over-the-air computation (AirComp) has been widely adopted for aggregating local gradients within SemiFL frameworks [13], [14]. As an analog transmission scheme, AirComp exploits the superposition property of wireless channels to directly aggregate devices' gradients during concurrent transmissions over the same time-frequency resources [15]. However, noise and fading in wireless channels cause over-the-air distortion, deviating the aggregated signal from its ideal form. Regarding this issue, prevailing approaches primarily adhere to the distortion-suppressing criterion that aims to minimize over-the-air distortion, typically requiring substantial transmit power to achieve [16]–[18]. Nevertheless, recent techniques such as gradient sparcification [19] and gradient compression [20] imply the robustness of gradients, suggesting that precise aggregation of local gradients may be overly conservative. Meanwhile, the work in [21] reports that adding artificial noise to gradients can improve the learning performance of AI models. Motivated by these observations, an

Jingheng Zheng, Hui Tian and Ping Zhang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhengjh@bupt.edu.cn; tianhui@bupt.edu.cn; pzhang@bupt.edu.cn).

Wanli Ni is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: niwanli@tsinghua.edu.cn).

Yang Tian is with the School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 102206, China (e-mail: tianyang9108@163.com).

intriguing yet critical question rises: *Is it possible to accelerate the convergence of SemiFL by intentionally leveraging over-the-air distortion during gradient aggregation*?

B. Related Work

AirComp has attracted attention in the field of both FL and SemiFL, such as the work in [9], [17], [18], [22]. Specifically, the authors of [17] mitigated wireless fading for FL model aggregation by integrating digital modulation with AirComp. In [18], the authors provided quantitative comparisons between a digital transmission scheme and AirComp, revealing the superior spectrum utilization of AirComp but its greater vulnerability to aggregation errors. As for SemiFL, the authors of [9] suppressed the over-the-air distortion of aggregated gradients by keeping the corresponding mean squared error (MSE) below a specific threshold. In [22], the authors aimed to minimize the latency of AirComp-based SemiFL, whereas over-the-air distortion was still suppressed by imposing an constraint on MSE. Although the above work uniformly dedicated to restricting over-the-air distortion, the authors of [23]–[25] have suggested that the noise of wireless channel could potentially ameliorate the convergence of FL. Thus, the above work is overly conservative in exploring the convergence acceleration effect of over-the-air distortion.

Other studies have been devoted to the joint allocation of multi-type resources in the SemiFL framework, including the work in [10], [26], [27]. Specifically, by jointly optimizing the communication and computation resources of a hybrid learning framework, the authors of [10] balanced model accuracy loss, latency, and energy consumption. Subject to restricted energy budget, the authors of [26] minimized the training latency of a hybrid learning framework by jointly optimizing device selection, the number of local and global iterations, and bandwidth allocation. The authors of [27] minimized the latency of a SemiFL framework comprising computing-heterogeneous devices by optimizing transmit power and the receive factor. Though the above work addresses the resource allocation issue to a limited degree, its scope has yet to encompass the idea of accelerating convergence by leveraging over-the-air distortion. In addition, the category of optimizable resources in the SemiFL framework can be further expanded to include more aspects such as the data allocation between devices and the BS.

C. Contributions and Organization

To address the aforementioned key problems, we develop a new SemiFL framework in this paper, which distinguishes itself from the existing SemiFL schemes proposed in [7]–[10] by replacing CL with split learning (SL) [28]. The new SemiFL framework can preserve data privacy by enabling devices to transmit only the intermediate outputs of the shallow layers of the global model rather than the raw data. Another important aspect is that we propose to accelerate the convergence of SemiFL by leveraging over-the-air distortion. In particular, by deriving a new MSE as a controllable metric of over-the-air distortion, we intentionally permit a high MSE threshold in the non-stable region to materialize convergence acceleration for SemiFL. Besides, theoretical results reveal the positive and negative roles of over-the-air distortion in different regions. With the aim of conserving energy, the antagonistic roles of over-the-air distortion necessitate different resource allocation schemes for different regions. The main contributions of this paper are summarized as follows:

- We propose a novel approach to accelerate the convergence of SemiFL through the strategic utilization of over-the-air distortion during gradient aggregation. Our approach triggers the acceleration effect by intentionally permitting a high MSE threshold while increasing the ratio of the power scaling factor to the normalizing factor in the non-stable region. To preserve devices' privacy, we resort to SL that splits the global model into two parts: devices handle the shallow layers, while the BS manages the deep layers. As such, devices process raw data with the shallow layers and upload only the intermediate outputs to the BS.
- We derive a closed-form lower bound for the expected loss reduction between two consecutive rounds in the non-stable region, revealing that high distortion helps accelerate convergence by boosting loss descent. This theoretically substantiates the acceleration effect of over-the-air distortion, while affirming that the ratio of the power scaling factor to the normalizing factor can efficiently modulate the degree of acceleration. However, we derive an upper bound of the optimality gap, which reveals an opposite effect of over-the-air distortion in the stable region: a large MSE increases the upper bound. Given the antagonistic roles of over-the-air distortion, we formulate two distinct problems to minimize energy consumption of SemiFL, each tailored to a specific type of region, subject to latency and resource constraints. We propose two resource allocation algorithms with closed-form solutions to solve the formulated problems by jointly optimizing communication and computation resources, as well as the data allocation between FL and SL.
- We evaluate the proposed framework and algorithms by using SemiFL to train a multilayer perceptron (MLP), a convolutional neural network (CNN), and a residual network (ResNet) on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, respectively. Simulation results validate that: 1) Over-the-air distortion effectively accelerates the convergence of SemiFL, surpassing the performance of state-of-the-art benchmarks. Specifically, it achieves accuracy gains of 4.47%, 18.39%, and 7.89% on the three datasets when compared to CL. 2) The ratio of the power scaling factor to the normalizing factor effectively modulates the acceleration effect of over-the-air distortion. 3) Energy consumption can be significantly reduced.

The remainder of this paper is organized as follows. In Section II, we describe the system model of SemiFL. In Section III, we present the theoretical analysis results and formulate two problems to reduce energy consumption. In Section IV, we present proposed algorithms, followed by simulation results in V. The paper is concluded in Section VI.

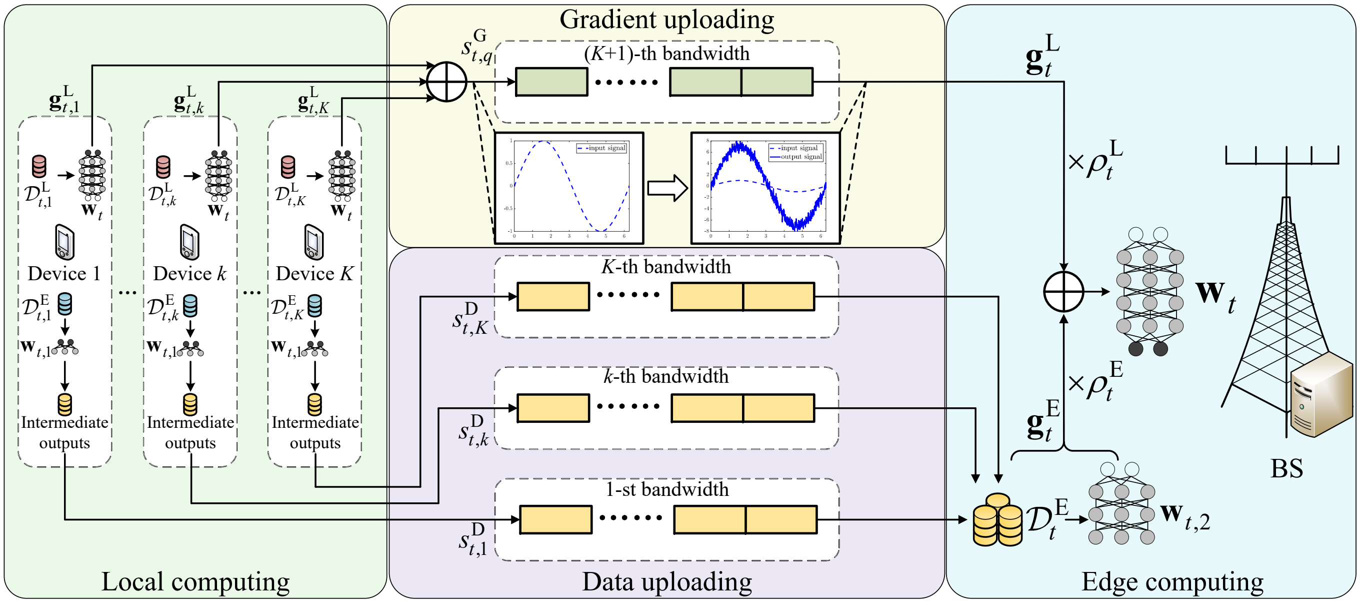

Gradient uploading
$s^{\mathrm{G}}_{t,q}$
(K+1)-th bandwidth
$\mathbf{g}^{\mathrm{L}}_{t}$
$\times\rho^{\mathrm{L}}_{t}$
$\mathbf{g}^{\mathrm{L}}_{t,1}$
$\mathbf{g}^{\mathrm{L}}_{t,k}$
$\mathbf{g}^{\mathrm{L}}_{t,K}$
$\mathcal{D}^{\mathrm{L}}_{t,1}$
$\mathcal{D}^{\mathrm{L}}_{t,k}$
$\mathcal{D}^{\mathrm{L}}_{t,K}$
$\mathbf{w}_t$
input signal
output signal
Device 1
Device k
Device K
$\mathcal{D}^{\mathrm{E}}_{t,1}$
$\mathcal{D}^{\mathrm{E}}_{t,k}$
$\mathcal{D}^{\mathrm{E}}_{t,K}$
$\mathbf{w}_{t,1}$
Intermediate outputs
K-th bandwidth
$s^{\mathrm{D}}_{t,K}$
k-th bandwidth
$s^{\mathrm{D}}_{t,k}$
1-st bandwidth
$s^{\mathrm{D}}_{t,1}$
$\mathbf{w}_t$
$\times\rho^{\mathrm{E}}_{t}$
$\mathbf{g}^{\mathrm{E}}_{t}$
BS
$\mathcal{D}^{\mathrm{E}}_{t}$
$\mathbf{w}_{t,2}$
Local computing
Data uploading
Edge computing

Fig. 1. An illustration of over-the-air distortion accelerated SemiFL.

II. System Model

As shown in Fig. 1, we consider a SemiFL framework consisting of an N_r-antenna BS and K single-antenna devices. The devices are indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$. The k-th device possess a local dataset, denoted by $\mathcal{D}_k$, containing $|\mathcal{D}_k| = D, \forall k \in \mathcal{K}$ data samples for training a shared global model $\mathbf{w} \in \mathbb{R}^Q$. Here, $|\cdot|$ denotes set cardinality, $\mathcal{R}$ denotes the set of real numbers, and Q denotes the dimensions of $\mathbf{w}$.

A. SemiFL Framework

Consider a T-round SemiFL where the round indexes are collected by the set $\mathcal{T} = \{1, 2, \ldots, T\}$. The goal is to minimize the global loss function $F(\mathbf{w})$, defined by

$$F(\mathbf{w}) = \sum\nolimits_{k=1}^{K} \sum\nolimits_{n=1}^{D} f(\mathbf{w}; \mathbf{\Omega}_{n,k}), \tag{1}$$

where $f(\mathbf{w}; \mathbf{\Omega}_{n,k})$ denotes the loss function regarding a data sample $\mathbf{\Omega}_{n,k}$. To this end, in the t-th round, the k-th device divides the local dataset $\mathcal{D}_k$ into two disjoint subsets, i.e., the edge dataset $\mathcal{D}^{\mathrm{E}}_{t,k}$ for SL at the BS containing $\theta_{t,k}D$ data samples and the local dataset $\mathcal{D}^{\mathrm{L}}_{t,k}$ consisting of $(1-\theta_{t,k})D$ data samples for local training, where $\theta_{t,k} \in (0, 1)$ denotes the ratio of SL data.

From the perspective of FL, the local dataset $\mathcal{D}^{\mathrm{L}}_{t,k}$ are retained locally for calculating the local gradient $\mathbf{g}^{\mathrm{L}}_{t,k} = [g^{\mathrm{L}}_{t,k,1}, \ldots, g^{\mathrm{L}}_{t,k,Q}]^{\mathrm{T}} \in \mathbb{R}^Q$, defined by

$$\mathbf{g}^{\mathrm{L}}_{t,k} = \sum_{n \in \mathcal{D}^{\mathrm{L}}_{t,k}} \nabla f(\mathbf{w}_t; \mathbf{\Omega}_{n,k}), \forall k \in \mathcal{K}, \tag{2}$$

where $\mathbf{w}_t$ denotes the global model in the t-th round and ∇ denotes the gradient operator. Then, all devices upload their local gradients $\{\mathbf{g}^{\mathrm{L}}_{t,k}\}$ to the BS for aggregation. To facilitate SL, we rewrite the global model $\mathbf{w}_t$ as a combination of shallow layers $\mathbf{w}_{t,1} \in \mathbb{R}^{Q_1}$ and deep layers $\mathbf{w}_{t,2} \in \mathbb{R}^{Q_2}$, i.e., $\mathbf{w}_t = [\mathbf{w}^{\mathrm{T}}_{t,1}, \mathbf{w}^{\mathrm{T}}_{t,2}]^{\mathrm{T}} \in \mathbb{R}^Q$ with $Q_1 + Q_2 = Q$. As a result, the aggregated gradient can be denoted by $\mathbf{g}^{\mathrm{L}}_t = [(\mathbf{g}^{\mathrm{L}}_{t,1})^{\mathrm{T}}, (\mathbf{g}^{\mathrm{L}}_{t,2})^{\mathrm{T}}]^{\mathrm{T}} = [g^{\mathrm{L}}_1, \ldots, g^{\mathrm{L}}_{Q_1}, g^{\mathrm{L}}_{Q_1+1}, \ldots, g^{\mathrm{L}}_Q]^{\mathrm{T}} \in \mathbb{R}^Q$, where $\mathbf{g}^{\mathrm{L}}_{t,1} \in \mathbb{R}^{Q_1}$ and $\mathbf{g}^{\mathrm{L}}_{t,2} \in \mathbb{R}^{Q_2}$ denote the aggregated gradients of shallow layers and deep layers, respectively.

To enable SL, the k-th device uploads the edge dataset $\mathcal{D}^{\mathrm{E}}_{t,k}$ to the BS. To preserve data privacy, the k-th device first inputs the raw data in $\mathcal{D}^{\mathrm{E}}_{t,k}$ into the shallow layers $\mathbf{w}_{t,1}$, and replace them with the resultant intermediate outputs. Then, the k-th device uploads these intermediate outputs in $\mathcal{D}^{\mathrm{E}}_{t,k}$ to the BS. The BS collects the intermediate outputs in a dataset $\mathcal{D}^{\mathrm{E}}_t$ and uses it to calculate the gradient of the deep layers $\mathbf{w}_{t,2}$. Concretely, the edge gradient, $\mathbf{g}^{\mathrm{E}}_t \in \mathbb{R}^{Q_2}$, is defined by

$$\mathbf{g}^{\mathrm{E}}_t = \sum_{n \in \mathcal{D}^{\mathrm{E}}_t} \nabla f(\mathbf{w}_{t,2}; \mathbf{\Omega}_n), \tag{3}$$

where $\mathbf{\Omega}_n$ denotes an intermediate output in $\mathcal{D}^{\mathrm{E}}_t$.

Using the aggregated gradient $\mathbf{g}^{\mathrm{L}}_t$ and the edge gradient $\mathbf{g}^{\mathrm{E}}_t$, the shallow layers $\mathbf{w}_{t,1}$ and deep layers $\mathbf{w}_{t,2}$ of global model $\mathbf{w}_t$ are updated by (4) and (5), respectively, given by

$$\mathbf{w}_{t+1,1} = \mathbf{w}_{t,1} - \eta_t \mathbf{g}^{\mathrm{L}}_{t,1}, \tag{4}$$

$$\mathbf{w}_{t+1,2} = \mathbf{w}_{t,2} - \eta_t(\rho^{\mathrm{L}}_t \mathbf{g}^{\mathrm{L}}_{t,2} + \rho^{\mathrm{E}}_t \mathbf{g}^{\mathrm{E}}_t), \tag{5}$$

where η_t denotes the learning rate. Here, $\rho^{\mathrm{L}}_t = 1 - (1/K)\sum_{k=1}^{K} \theta_{t,k}$ and $\rho^{\mathrm{E}}_t = (1/K)\sum_{k=1}^{K} \theta_{t,k}$ denote the FL and SL weight coefficients, respectively. Next, the BS broadcasts the updated global model $\mathbf{w}_{t+1} = [\mathbf{w}^{\mathrm{T}}_{t+1,1}, \mathbf{w}^{\mathrm{T}}_{t+1,2}]^{\mathrm{T}}$ to all devices for the next round of SemiFL.

B. Over-the-Air Gradient Aggregation and Data Uploading

As depicted in Fig. 1, $K+1$ orthogonal bandwidths are employed to aggregate local gradients $\{\mathbf{g}^{\mathrm{L}}_{t,k}\}$ over the air while uploading datasets $\{\mathcal{D}^{\mathrm{E}}_{t,k}\}$. The k-th device occupies a dedicated bandwidth to upload the intermediate outputs in $\mathcal{D}^{\mathrm{E}}_{t,k}$. The remaining bandwidth is shared by all K devices to aggregate local gradients $\{\mathbf{g}^{\mathrm{L}}_{t,k}\}$ using AirComp.

1) Over-the-Air Gradient Aggregation: Let $\hat{\mathbf{g}}^{\mathrm{L}}_{t,k} = [\hat{g}^{\mathrm{L}}_{t,k,1}, \ldots, \hat{g}^{\mathrm{L}}_{t,k,Q}] \in \mathbb{R}^{Q}$ denote the normalized gradient signal of the local gradient $\mathbf{g}^{\mathrm{L}}_{t,k}$, where $\mathbb{E}[\hat{g}^{\mathrm{L}}_{t,k,q}] = 0$ and $\mathbb{E}[|\hat{g}^{\mathrm{L}}_{t,k,q}|^2] = 1, \forall q$. Devices concurrently send gradient signals $\{\hat{\mathbf{g}}^{\mathrm{L}}_{t,k}\}$ over the shared bandwidth entry by entry. Then, the q-th signal of the gradient aggregated over the air is

$$s^{\mathrm{G}}_{t,q} = \frac{\mathbf{b}^{\mathrm{H}}_t}{K\sqrt{\nu_t}} \left(\sum\nolimits_{k=1}^{K} \mathbf{h}^{\mathrm{G}}_{t,k} p^{\mathrm{G}}_{t,k} \hat{g}^{\mathrm{L}}_{t,k,q} + \mathbf{n}^{\mathrm{G}}_t \right), \forall q, \tag{6}$$

where $\mathbf{b}_t \in \mathbb{C}^{N_r}$ satisfying $\|\mathbf{b}_t\| = 1$ and $\nu_t > 0$ denote the receive beamformer and the normalizing factor for gradient aggregation, respectively, $\mathbf{h}^{\mathrm{G}}_{t,k} \in \mathbb{C}^{N_r}$ denotes the channel coefficient vector between the k-th device and the BS for gradient aggregation, $p^{\mathrm{G}}_{t,k}$ denotes the transmit power coefficient of the k-th device for gradient uploading, and $\mathbf{n}^{\mathrm{G}}_t \in \mathbb{C}^{N_r}$ yielding $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ denotes the noise. Here, $(\cdot)^{\mathrm{H}}$, $\|\cdot\|$, σ^2, and $\mathbf{I}$ denote conjugate transpose, vector 2-norm, the noise power, and an identity matrix, respectively. We set the transmit power coefficient of the k-th device for gradient uploading to $p^{\mathrm{G}}_{t,k} = \sqrt{\omega_t}(\mathbf{b}^{\mathrm{H}}_t\mathbf{h}^{\mathrm{G}}_{t,k})^{\mathrm{H}}/|\mathbf{b}^{\mathrm{H}}_t\mathbf{h}^{\mathrm{G}}_{t,k}|^2$, where $\omega_t > 0$ denotes the power scaling factor. As a result, the q-th entry of the aggregated gradient $\mathbf{g}^{\mathrm{L}}_t$ can be obtained by

$$g^{\mathrm{L}}_{t,q} = \mathrm{Re}\{s^{\mathrm{G}}_{t,q}\} = \frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} \frac{1}{K} \sum\nolimits_{k=1}^{K} \hat{g}^{\mathrm{L}}_{t,k,q} + \hat{n}^{\mathrm{G}}_{t,q}, \forall q, \tag{7}$$

where $\mathrm{Re}\{\cdot\}$ takes the real part and $\hat{n}^{\mathrm{G}}_{t,q} = \mathrm{Re}\{\mathbf{b}^{\mathrm{H}}_t\mathbf{n}^{\mathrm{G}}_t\}/\sqrt{\nu_t}$, yielding $\mathcal{N}(0, \frac{\sigma^2}{2\nu_t})$.

Denote the ideally aggregated gradient by $\hat{\mathbf{g}}^{\mathrm{L}}_t = [\hat{g}^{\mathrm{L}}_{t,1}, \ldots, \hat{g}^{\mathrm{L}}_{t,Q}] \in \mathbb{R}^{Q}$, where $\hat{g}^{\mathrm{L}}_{t,q} = (\sum_{k=1}^{K} \hat{g}^{\mathrm{L}}_{t,k,q})/K, \forall q$. Then, by assembling the received entries, the aggregated gradient $\mathbf{g}^{\mathrm{L}}_t$ can be expressed by

$$\mathbf{g}^{\mathrm{L}}_t = \frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} \hat{\mathbf{g}}^{\mathrm{L}}_t + \hat{\mathbf{n}}^{\mathrm{G}}_t, \tag{8}$$

where $\hat{\mathbf{n}}^{\mathrm{G}}_t = [\hat{n}^{\mathrm{G}}_{t,1}, \ldots, \hat{n}^{\mathrm{G}}_{t,Q}] \in \mathbb{R}^{Q}$, yielding $\mathcal{N}(\mathbf{0}, \frac{\sigma^2}{2\nu_t}\mathbf{I})$. Meanwhile, the MSE between $g^{\mathrm{L}}_{t,q}$ and $\hat{g}^{\mathrm{L}}_{t,q}$, i.e., $\mathrm{MSE}_t = \mathbb{E}[|g^{\mathrm{L}}_{t,q} - \hat{g}^{\mathrm{L}}_{t,q}|^2]$, is used as a metric to quantify the over-the-air distortion, as derived by

$$\mathrm{MSE}_t = \frac{1}{K}\left(\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} - 1\right)^2 + \frac{\sigma^2}{2\nu_t}. \tag{9}$$

2) Data Uploading: Apart from gradient aggregation, the k-th device also sends the intermediate outputs in $\mathcal{D}^{\mathrm{E}}_{t,k}$ to the BS on its dedicated bandwidth. Denote the signal of the intermediate outputs from the k-th device by $d_{t,k}$, where $\mathbb{E}[d_{t,k}] = 0$ and $\mathbb{E}[|d_{t,k}|^2] = 1$. Then, the data signal of the k-th device received by the BS, $s^{\mathrm{D}}_{t,k}$, can be given by

$$s^{\mathrm{D}}_{t,k} = \frac{\mathbf{v}^{\mathrm{H}}_{t,k}}{\sqrt{\zeta_{t,k}}} \left(\mathbf{h}^{\mathrm{D}}_{t,k} p^{\mathrm{D}}_{t,k} d_{t,k} + \mathbf{n}^{\mathrm{D}}_t\right), \forall k \in \mathcal{K}, \tag{10}$$

where $\mathbf{v}^{\mathrm{H}}_{t,k}$ satisfying $\|\mathbf{v}^{\mathrm{H}}_{t,k}\| = 1$ and $\zeta_{t,k} > 0$ denote the receive beamformer and the normalizing factor for data uploading, respectively, $\mathbf{h}^{\mathrm{D}}_{t,k}$ denotes the channel coefficient vector between the k-th device and the BS for data uploading, $p^{\mathrm{D}}_{t,k}$ denotes the transmit power of the k-th device for data uploading, and $\mathbf{n}^{\mathrm{D}}_t$ yielding $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ denotes the noise. We set the transmit power of the k-th deice for data uploading to $p^{\mathrm{D}}_{t,k} = \sqrt{\zeta_{t,k}}(\mathbf{v}^{\mathrm{H}}_{t,k}\mathbf{h}^{\mathrm{D}}_{t,k})^{\mathrm{H}}/|\mathbf{v}^{\mathrm{H}}_{t,k}\mathbf{h}^{\mathrm{D}}_{t,k}|^2$. Then, the data signal $s^{\mathrm{D}}_{t,k}$ is reduced to

$$s^{\mathrm{D}}_{t,k} = d_{t,k} + \frac{\mathbf{v}^{\mathrm{H}}_{t,k}\mathbf{n}^{\mathrm{D}}_t}{\sqrt{\zeta_{t,k}}}, \forall k \in \mathcal{K}. \tag{11}$$

Based on (11), one can calculate the signal-to-noise ratio (SNR) and achievable data rate of the k-th device via (12) and (13), respectively, given by

$$\mathrm{SNR}_{t,k} = \frac{\zeta_{t,k}}{\|\mathbf{v}_{t,k}\|^2\sigma^2} = \frac{\zeta_{t,k}}{\sigma^2}, \forall k \in \mathcal{K}, \tag{12}$$

$$R_{t,k} = B\log\left(1 + \frac{\zeta_{t,k}}{\sigma^2}\right), \forall k \in \mathcal{K}, \tag{13}$$

where B denotes the bandwidth of each spectrum segment for data uploading and $\log(\cdot)$ denotes the natural logarithm.

C. Latency and Energy Consumption

The t-th round of the SemiFL process is mainly composed of four phases, including the gradient uploading phase, the data uploading phase, the local computing phase, and the edge computing phase [29]. Note that we ignore the model broadcasting from the BS to devices, as the downlink transmit power of the BS is much higher than devices, and the BS can access to sufficient energy supply [30], [31]. The latency and energy consumption of each phase are modeled as follows:

1) Gradient Uploading: Suppose each AirComp block contains M gradient signals. The latency of gradient uploading, T^{G}_t, is given by

$$T^{\mathrm{G}}_t = \left\lceil \frac{Q}{M} \right\rceil T_s, \tag{14}$$

where T_s denotes the duration of an AirComp block and $\lceil\cdot\rceil$ denotes the ceiling function. The energy consumption of the k-th device for uploading gradient can be given by

$$E^{\mathrm{G}}_{t,k} = |p^{\mathrm{G}}_{t,k}|^2 T^{\mathrm{G}}_t = \frac{\omega_t \lceil \frac{Q}{M} \rceil T_s}{|\mathbf{b}^{\mathrm{H}}_t\mathbf{h}^{\mathrm{G}}_{t,k}|^2}, \forall k \in \mathcal{K}. \tag{15}$$

2) Data Uploading: Suppose an intermediate output is presented by $\bar{C}$ bits. Then, the data uploading latency of the k-th device, $T^{\mathrm{D}}_{t,k}$, can be calculated by

$$T^{\mathrm{D}}_{t,k} = \frac{D\theta_{t,k}\bar{C}}{R_{t,k}}, \forall k \in \mathcal{K}. \tag{16}$$

Consequently, the energy consumption of the k-th device for data uploading can be given by

$$E^{\mathrm{D}}_{t,k} = |p^{\mathrm{D}}_{t,k}|^2 T^{\mathrm{D}}_{t,k} = \frac{\zeta_{t,k} D\theta_{t,k}\bar{C}}{|\mathbf{v}^{\mathrm{H}}_{t,k}\mathbf{h}^{\mathrm{D}}_{t,k}|^2 R_{t,k}}, \forall k \in \mathcal{K}. \tag{17}$$

3) Local Computing: Denote the CPU frequency of the k-th device by $\hat{f}_{t,k}$. Then, the local computing latency of the k-th device, $T^{\mathrm{L}}_{t,k}$, is given by

$$T^{\mathrm{F}}_{t,k} = \frac{D(1-\theta_{t,k})\hat{C}_k}{\hat{f}_{t,k}}, \forall k \in \mathcal{K}, \tag{18}$$

where $\hat{C}_k$ denotes the number of CPU circles required for a data sample. The energy consumption of the k-th device for local computing is given by [32]

$$E_{t,k}^{\mathrm{F}} = D(1-\theta_{t,k})\hat{C}_k \hat{\kappa} \hat{f}_{t,k}^2, \forall k \in \mathcal{K}, \tag{19}$$

where $\hat{\kappa}$ denotes the effective switched capacitance of devices.

4) Edge Computing: Denoting the CPU frequency of the BS by $\tilde{f}_t$, the edge computing latency of the BS is given by

$$T_t^{\mathrm{E}} = \frac{D(\sum_{k=1}^{K} \theta_{t,k})\tilde{C}}{\tilde{f}_t}, \tag{20}$$

where $D(\sum_{k=1}^{K} \theta_{t,k})$ denotes the number of intermediate outputs in $\mathcal{D}_t^{\mathrm{E}}$, and $\tilde{C}$ denotes the number of CPU circles for processing an intermediate output. The energy consumption of the BS for edge computing is given by

$$E_t^{\mathrm{E}} = D\left(\sum\nolimits_{k=1}^{K} \theta_{t,k}\right)\tilde{C}\tilde{\kappa}\tilde{f}_t^2, \tag{21}$$

where $\tilde{\kappa}$ denotes the effective switched capacitance of the BS.

5) Overall Latency and Energy Consumption: Based on (14)–(21), the overall latency of the t-th round is given by

$$\begin{aligned} T_t^{\mathrm{ALL}} = \max\{&T_{t,1}^{\mathrm{D}} + T_t^{\mathrm{E}}, \dots, T_{t,K}^{\mathrm{D}} + T_t^{\mathrm{E}}, \\ &T_{t,1}^{\mathrm{F}} + T_t^{\mathrm{G}}, \dots, T_{t,K}^{\mathrm{F}} + T_t^{\mathrm{G}}\}. \end{aligned} \tag{22}$$

In addition, the overall energy consumption of the t-th round can be calculated by

$$E_t^{\mathrm{ALL}} = \sum\nolimits_{k=1}^{K} (E_{t,k}^{\mathrm{G}} + E_{t,k}^{\mathrm{D}} + E_{t,k}^{\mathrm{F}}) + E_t^{\mathrm{E}}. \tag{23}$$

III. Convergence Analysis and Problem Formulation

In this section, as shown in Fig. 2, the SemiFL process is categorized into two types of regions: the non-stable region $\mathcal{R}^{\mathrm{NS}}$ and the stable region $\mathcal{R}^{\mathrm{S}}$, which are defined by $\mathcal{R}^{\mathrm{NS}} = \{\mathbf{w}_t | \, \|\nabla F(\mathbf{w}_t)\| \geq \varepsilon, \forall t \in \mathcal{T}\}$ and $\mathcal{R}^{\mathrm{S}} = \{\mathbf{w}_t | \, \|\nabla F(\mathbf{w}_t)\| < \varepsilon, \forall t \in \mathcal{T}\}$ [25], respectively, where the constant $\varepsilon > 0$. Next, we analyze the convergence of SemiFL, revealing the antagonistic roles of over-the-air distortion in different regions. In view of the opposing effects of over-the-air distortion, we propose a two-region MSE threshold configuration scheme, whereby two distinct problems are formulated for each type of region to minimize energy consumption.

A. Convergence Analysis

Assumption 1. *The global loss function $F(\mathbf{w})$ is L-smooth regarding a constant $L > 0$. For any $\mathbf{w}$, $\mathbf{w}' \in \mathbb{R}^Q$, we have*

$$F(\mathbf{w}) \leq F(\mathbf{w}') + \nabla F(\mathbf{w}')^{\mathrm{T}} (\mathbf{w} - \mathbf{w}') + \frac{L}{2}\|\mathbf{w} - \mathbf{w}'\|^2. \tag{24}$$

Assumption 2. *The global loss function $F(\mathbf{w})$ is μ-strongly convex. For any $\mathbf{w}$, $\mathbf{w}' \in \mathbb{R}^Q$ and a constant $\mu > 0$, we have*

$$F(\mathbf{w}) \geq F(\mathbf{w}') + \nabla F(\mathbf{w}')^{\mathrm{T}} (\mathbf{w} - \mathbf{w}') + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}'\|^2. \tag{25}$$

Assumption 3. *The expected squared 2-norm of the global gradient $\nabla F(\mathbf{w}_t)$ is bounded by a constant $A^2 > 0$. For any $\mathbf{w}_t \in \mathbb{R}^Q$, we have*

$$\mathbb{E}[\|\nabla F(\mathbf{w}_t)\|^2] \leq A^2. \tag{26}$$

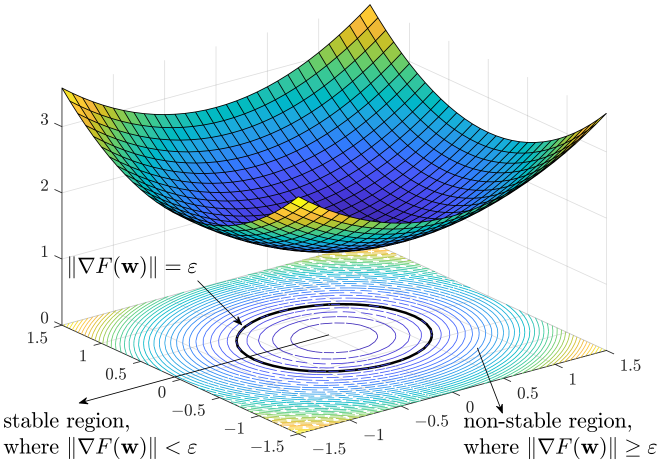


Fig. 2. An illustration of the non-stable and stable regions of SemiFL.

Assumption 4. *The ideally aggregated gradient $\hat{\mathbf{g}}_t^{\mathrm{L}}$ and the edge gradient $\mathbf{g}_t^{\mathrm{E}}$ are unbiased estimations of the global gradient $\nabla F(\mathbf{w}_t)$. Thus, we have*

$$\mathbb{E}[\hat{\mathbf{g}}_t^{\mathrm{L}}] = \nabla F(\mathbf{w}_t), \tag{27}$$

$$\mathbb{E}[\mathbf{g}_t^{\mathrm{E}}] = \nabla F(\mathbf{w}_t). \tag{28}$$

We are now in position to characterize the convergence behavior of SemiFL in the non-stable region $\mathcal{R}^{\mathrm{NS}}$. Specifically, we derive a closed-form lower bound for the expected loss function reduction between two consecutive rounds in $\mathcal{R}^{\mathrm{NS}}$, as presented in Theorem 1.

Theorem 1. *Given Assumptions 2 – 4, for global models $\mathbf{w}_t$, $\mathbf{w}_{t+1} \in \mathcal{R}^{\mathrm{NS}}$ and $\varepsilon \geq A/\sqrt{2\mu}$, the expected global loss function reduction between two consecutive rounds in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ is lower bounded by*

$$\begin{aligned} \mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}_{t+1})] \geq \frac{\eta_t^2}{4}(2\mu\varepsilon^2 - A^2)\left[1 + \left(\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} - 1\right)\rho_t^{\mathrm{L}}\right]^2 \\ - A^2 + \frac{\mu\sigma^2 Q \eta_t^2}{4\nu_t}(\rho_t^{\mathrm{L}})^2. \end{aligned} \tag{29}$$

Proof: Please refer to Appendix B in [33]. □

Remark 1. *Based on the lower bound (29) in Theorem 1, we have the following two key insights in terms of the convergence acceleration effect of over-the-air distortion for SemiFL:*

- *Recalling the definition of MSE_t in (9), it is seen that either an increase in the power scaling factor-to-normalizing factor ratio $\frac{\sqrt{\omega_t}}{\sqrt{\nu}_t}$ or a decrease in the normalizing factor ν_t leads to a higher MSE_t, implying amplified over-the-air distortion. However, based on (29), both an increasing $\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}}$ and a decreasing ν_t increase the lower bound of $\mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}_{t+1})]$. Hence, this offers the insight that: by setting a higher MSE threshold, one can adopt a higher $\frac{\sqrt{\omega_t}}{\sqrt{\nu}_t}$ and a lower ν_t to amplify over-the-air distortion, thereby accelerating the convergence, i.e., loss descent, of SemiFL in the non-stable region $\mathcal{R}^{\mathrm{NS}}$.*
- *For $\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} \geq 1$, it is observed in (29) that the lower bound of $\mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}_{t+1})]$ increases with the FL weight coefficient ρ_t^{L}. This suggests another insight: the convergence acceleration effect of over-the-air distortion*

can be enhanced by uploading fewer data to the BS in the non-stable region $\mathcal{R}^{\mathrm{NS}}$, as ρ_t^{L} can be increased to boost the loss descent in SemiFL.

Then, in the stable region $\mathcal{R}^{\mathrm{S}}$, we derive an upper bound for the expected optimality gap of the global loss function $F(\mathbf{w}_t)$ as t approaches infinity, which is detailed in Theorem 2.

Theorem 2. *Given Assumptions 1 – 4, set $\eta_t = \frac{1}{\mu}$, $\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} = 1$, and $\nu_t = \nu, \forall t \in \mathcal{T}$. For a global model $\mathbf{w}_t \in \mathcal{R}^{\mathrm{S}}$, as $t \to \infty$, the expected optimality gap of the global loss function can be upper bounded by*

$$\lim_{t\to\infty} \mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)] \leq \frac{L}{\mu}\frac{1}{4\mu - L}\left(A^2 + \frac{\sigma^2 Q}{2\nu}\right), \quad (30)$$
$$\triangleq \psi(\nu),$$

where $\mathbf{w}^*$ denotes the optimal global model.

Proof: Please refer to Appendix C in [33]. □

Remark 2. *Based on Theorem 2, the smaller $\psi(\nu)$ is, the better the final convergence of SemiFL will be. Based on (30), one can find that $\psi(\nu)$ decreases as ν increases. Meanwhile, if $\frac{\sqrt{\omega_t}}{\sqrt{\nu_t}} = 1$, an increase in ν contributes to decreasing MSE_t. Therefore, achieving improved final convergence of SemiFL requires reducing MSE_t in the stable region $\mathcal{R}^{\mathrm{S}}$, or equivalently, suppressing over-the-air distortion.*

By comparing Remarks 1 and 2, it is observed that the over-the-air distortion plays antagonistic roles in different regions. Specifically, amplifying over-the-air distortion in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ accelerates the convergence of SemiFL, while suppressing over-the-air distortion in the stable region $\mathcal{R}^{\mathrm{S}}$ improves final convergence. The opposing effects of over-the-air distortion motivate the exploration of a two-region MSE threshold configuration scheme to leverage advantages from both sides, as elaborated in Remark 3.

Remark 3 (Two-region MSE threshold configuration.)**.** *The key idea of this scheme lies in setting a high MSE threshold in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ and a low MSE threshold in the stable region $\mathcal{R}^{\mathrm{S}}$. As such, a low ν_t can be adopted in $\mathcal{R}^{\mathrm{NS}}$, while a high ν_t can be applied in $\mathcal{R}^{\mathrm{S}}$, i.e., $\nu_t = \nu^{\mathrm{low}}, \forall t \in \{t|\ \mathbf{w}_t \in \mathcal{R}^{\mathrm{NS}}\}$ and $\nu_t = \nu^{\mathrm{high}}, \forall t \in \{t|\ \mathbf{w}_t \in \mathcal{R}^{\mathrm{S}}\}$, where $\nu^{\mathrm{high}} \geq \nu^{\mathrm{low}}$. Suppose SemiFL reaches the stable region $\mathcal{R}^{\mathrm{S}}$ in the T'-th round. For $t \geq T'$, we have*

$$\lim_{t\to\infty} \mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)]$$
$$\leq \lim_{t\to\infty} \{\xi^{t-1}\mathbb{E}[F(\mathbf{w}_1) - F(\mathbf{w}^*)] + \frac{L}{2\mu^2}\sum_{\tau=T'}^{t-1}\xi^{-\tau}(A^2 + \frac{\sigma^2 Q}{2\nu^{\mathrm{high}}})\}$$
$$\underbrace{+\lim_{t\to\infty}\frac{L}{2\mu^2}\sum\nolimits_{\tau=1}^{T'-1}\xi^{-\tau}(A^2 + \frac{\sigma^2 Q}{2\nu^{\mathrm{low}}})}_{\textit{negative effect of amplified over-the-air distortion in } \mathcal{R}^{\mathrm{S}}}$$
$$= \frac{L}{\mu}\frac{1}{4\mu - L}(A^2 + \frac{\sigma^2 Q}{2\nu^{\mathrm{high}}}). \quad (31)$$

where $\xi = L/(2\mu) - 1$. Inequality (31) demonstrates that the two-region MSE threshold configuration scheme not only leverages amplified over-the-air distortion in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ to accelerate convergence, but also mitigates its negative effect gradually in the stable region $\mathcal{R}^{\mathrm{S}}$ to achieve improved final convergence for SemiFL.

B. Problem Formulation

Following the two-region MSE threshold configuration scheme, we aim to minimize the per-round energy consumption in different types of regions by jointly optimizing the normalizing factors, power scaling factor, receive beamformers, CPU frequencies, and data allocation.

1) Non-Stable Region: The optimization problem of the t-th round, where $t \in \{t|\ \mathbf{w}_t \in \mathcal{R}^{\mathrm{NS}}\}$, is formulated as

$$\begin{aligned}
\min_{\substack{\{\zeta_k\},\nu,\omega,\mathbf{b},\{\mathbf{v}_k\},\\ \{\hat{f}_k\},\tilde{f},\{\theta_k\}}} \quad & E^{\mathrm{ALL}} && \text{(32a)}\\
\text{s.t.} \quad & T^{\mathrm{ALL}} \leq T_{\max}, && \text{(32b)}\\
& \epsilon_1 = \frac{\sqrt{\omega}}{\sqrt{\nu}}, && \text{(32c)}\\
& \mathrm{MSE} \leq \epsilon_2, && \text{(32d)}\\
& 0 \leq \zeta_k \leq p_{\max}|\mathbf{v}_k^{\mathrm{H}}\mathbf{h}_k^{\mathrm{D}}|^2, \forall k \in \mathcal{K}, && \text{(32e)}\\
& 0 \leq \omega \leq p_{\max}|\mathbf{b}^{\mathrm{H}}\mathbf{h}_k^{\mathrm{G}}|^2, \forall k \in \mathcal{K}, && \text{(32f)}\\
& 0 \leq \theta_k \leq \theta_{\max}, \forall k \in \mathcal{K}, && \text{(32g)}\\
& 0 \leq \hat{f}_k \leq \hat{f}_{\max}, \forall k \in \mathcal{K}, && \text{(32h)}\\
& 0 \leq \tilde{f} \leq \tilde{f}_{\max}, && \text{(32i)}\\
& \|\mathbf{v}_k\| = 1, \forall k \in \mathcal{K}, && \text{(32j)}\\
& \|\mathbf{b}\| = 1, && \text{(32k)}
\end{aligned}$$

where $T_{\max}$ denotes the maximum allowable latency per round, $\epsilon_1 > 1$ denotes the minimum power scaling factor-to-normalizing factor ratio in $\mathcal{R}^{\mathrm{NS}}$, $\epsilon_2 > 0$ denotes the MSE threshold in $\mathcal{R}^{\mathrm{NS}}$, $p_{\max}$ denotes the maximum transmit power of devices, $\theta_{\max}$ denotes the maximum ratio of SL data, $\hat{f}_{\max}$ and $\tilde{f}_{\max}$ denote the maximum CPU frequencies of devices and the BS, respectively. The subscript t is omitted. As we intend to accelerate the convergence of SemiFL in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ by amplifying over-the-air distortion, the ratio $\frac{\sqrt{\omega}}{\sqrt{\nu}}$ is forced to be greater than 1 in constraint (32c). Correspondingly, we choose a high MSE threshold ϵ_2 in constraint (32d). This aligns with the discussions in Remark 1. Moreover, we guarantee a large FL weight coefficient ρ_t^{L} by intentionally imposing a maximum allowable ratio of SL data, $\theta_{\max}$, so as to better utilize the acceleration effect of over-the-air distortion, as also discussed in Remark 1.

2) Stable Region: The optimization problem of the t-th round, where $t \in \{t|\ \mathbf{w}_t \in \mathcal{R}^{\mathrm{S}}\}$, is formulated by

$$\begin{aligned}
\min_{\substack{\{\zeta_k\},\nu,\omega,\mathbf{b},\{\mathbf{v}_k\},\\ \{\hat{f}_k\},\tilde{f},\{\theta_k\}}} \quad & E^{\mathrm{ALL}} && \text{(33a)}\\
\text{s.t.} \quad & \psi(\nu) \leq \epsilon_3, && \text{(33b)}\\
& \frac{\sqrt{\omega}}{\sqrt{\nu}} = 1, && \text{(33c)}\\
& \mathrm{MSE} \leq \epsilon_4, && \text{(33d)}\\
& \theta_{\min} \leq \theta_k \leq 1, \forall k \in \mathcal{K}, && \text{(33e)}\\
& \text{(32b), (32e), (32f), and (32h) – (32k),} && \text{(33f)}
\end{aligned}$$

where ϵ_3 denotes the maximum expected optimality gap, ϵ_4 denotes the MSE threshold in $\mathcal{R}^{\mathrm{S}}$, $\theta_{\min}$ denotes the minimum ratio of SL data. The subscript t is omitted. Note that we restrict the expected optimality gap to be below ϵ_3 in constraint (33b) to guarantee the final convergence of SemiFL. To mitigate the negative effect of over-the-air distortion $\mathcal{R}^{\mathrm{S}}$, we impose a low MSE threshold $\epsilon_4 \ll \epsilon_2$ in constraint (33d), as analyzed in Remark 3. In addition, imposing constraint (33c) helps suppress over-the-air distortion by forcing the first term of MSE to be zero according to (9).

Due to the indefinite Hessian matrices of E^{ALL}, T^{ALL}, and MSE, problems (32) and (33) are non-convex and intractable. In the next section, we propose two algorithms to solve the formulated two problems.

IV. Proposed Algorithms

In this section, we decompose problem (32) into four subproblems. By solving each problem iteratively, we obtain a solution to problem (32), with closed-form solutions derived for some subproblems. Then, we solve problem (33) in a similar manner.

A. Algorithm for Non-Stable Region

We first decompose problem (32) in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ into four subproblems and conquer them one by one, as presented below:

1) Normalizing and Power Scaling Factors: Given receive beamformers, CPU frequencies, and data allocation, the problem of normalizing and power scaling factors is reduced to

$$\begin{aligned}
\min_{\{\zeta_k\},\nu,\omega,\tau_1} \quad & \sum\nolimits_{k=1}^{K} \left[C_{1,k} \frac{\zeta_k}{\log(1+\frac{\zeta_k}{\sigma^2})} + C_{2,k}\omega \right] && \text{(34a)}\\
\text{s.t.} \quad & \tau_1 - T_{\max} \le 0, && \text{(34b)}\\
& \frac{C_{3,k}}{\log(1+\frac{\zeta_k}{\sigma^2})} + T^{\mathrm{E}} - \tau_1 \le 0, \forall k \in \mathcal{K}, && \text{(34c)}\\
& \max_{k\in\mathcal{K}}\{T_k^{\mathrm{L}}\} + T^{\mathrm{G}} - \tau_1 \le 0, && \text{(34d)}\\
& \zeta_k - C_{4,k} \le 0, \forall k \in \mathcal{K}, && \text{(34e)}\\
& \omega - C_5 \le 0, && \text{(34f)}\\
& C_6\sqrt{\nu} - \sqrt{\omega} = 0, && \text{(34g)}\\
& C_7\nu - 2\sqrt{\omega}\sqrt{\nu} + \omega + C_8 \le 0, && \text{(34h)}
\end{aligned}$$

where $C_{1,k} = D\bar{C}\theta_k/(|\mathbf{v}_k^{\mathrm{H}}\mathbf{h}_k^{\mathrm{D}}|^2 B), \forall k \in \mathcal{K}$, $C_{2,k} = \lceil Q/M \rceil T_s/|\mathbf{b}^{\mathrm{H}}\mathbf{h}_k^{\mathrm{G}}|^2, \forall k \in \mathcal{K}$, $C_{3,k} = D\bar{C}\theta_k/B, \forall k \in \mathcal{K}$, $C_{4,k} = p_{\max}|\mathbf{v}_k^{\mathrm{H}}\mathbf{h}_k^{\mathrm{D}}|^2, \forall k \in \mathcal{K}$, $C_5 = p_{\max}\min_{k\in\mathcal{K}}|\mathbf{b}^{\mathrm{H}}\mathbf{h}_k^{\mathrm{G}}|^2$, $C_6 = \epsilon_1$, $C_7 = 1 - K\epsilon_2$, $C_8 = K\sigma^2/2$, and $\tau_1 \ge T^{\mathrm{ALL}}$ is an auxiliary variable. Problem (34) is non-convex because of the existence of the concave term $\zeta_k/\log(1+\zeta_k/\sigma^2)$. To make problem (34) tractable, we further decompose it into two subproblems: one regarding the normalizing factors $\{\zeta_k\}$ and τ_1, and the other regarding the normalizing factors ν and ω.

By plugging in $\sqrt{\omega} = C_6\sqrt{\nu}$ based on constraint (34g), the subproblem regarding ν and ω is reduced to

$$\begin{aligned}
\min_{\nu} \quad & \left(\sum\nolimits_{k=1}^{K} C_{2,k}\right) C_6^2 \nu && \text{(35a)}\\
\text{s.t.} \quad & -\frac{C_8}{C_7 - 2C_6 + C_6^2} \le \nu \le \frac{C_5}{C_6}. && \text{(35b)}
\end{aligned}$$

Since $(\sum_{k=1}^{K} C_{2,k})C_6^2 \ge 0$, the closed-form solution to problem (35) is given by

$$\nu^* = -\frac{C_8}{C_7 - 2C_6 + C_6^2}. \tag{36}$$

Correspondingly, the closed-form solution to ω is given by

$$\omega^* = -\frac{C_6^2 C_8}{C_7 - 2C_6 + C_6^2}. \tag{37}$$

On the other hand, for notation convenience, define two functions: $h_{1,k}(\zeta_k) = \zeta_k/\log(1+\zeta_k/\sigma^2), \forall k \in \mathcal{K}$ and $h_{2,k}(\tau_1) = \sigma^2(2^{C_{3,k}/(\tau_1 - T^{\mathrm{E}})} - 1), \forall k \in \mathcal{K}$. Then, the subproblem regarding $\{\zeta_k\}$ and τ_1 is given by

$$\begin{aligned}
\min_{\{\zeta_k\},\tau_1} \quad & \sum\nolimits_{k=1}^{K} C_{1,k} h_{1,k}(\zeta_k) && \text{(38a)}\\
\text{s.t.} \quad & \max_{k\in\mathcal{K}}\{T_k^{\mathrm{L}}\} + T^{\mathrm{G}} \le \tau_1 \le T_{\max}, && \text{(38b)}\\
& h_{2,k}(\tau_1) \le \zeta_k \le C_{4,k}, \forall k \in \mathcal{K}, && \text{(38c)}
\end{aligned}$$

which is non-convex because of the non-convexity of (38a). However, it is noticed that $h_{1,k}(\zeta_k)$ is a monotonically increasing univariate concave function of ζ_k whose minimum can be attained at the boundary of the feasible region [34]. Furthermore, since $h_{2,k}(\tau_1)$ decreases as τ_1 increases, τ_1 should be maximized within the feasible region of problem (38) to minimize $h_{2,k}(\tau_1)$. Thus, the optimal τ_1 is given by

$$\tau_1^* = T_{\max}. \tag{39}$$

Then, to minimize (38a), one should minimize ζ_k within the feasible region. Hence, the closed-form optimal normalizing factors $\{\zeta_k\}$ are given by

$$\zeta_k^* = h_{2,k}(T_{\max}), \forall k \in \mathcal{K}. \tag{40}$$

2) Receive Beamformers: For notation convenience, define two rank-one matrices: $\mathbf{V}_k = [\mathrm{Re}\{\mathbf{v}_k\}^{\mathrm{T}}, \mathrm{Im}\{\mathbf{v}_k\}^{\mathrm{T}}]^{\mathrm{T}}[\mathrm{Re}\{\mathbf{v}_k\}^{\mathrm{T}}, \mathrm{Im}\{\mathbf{v}_k\}^{\mathrm{T}}], \forall k \in \mathcal{K}$ and $\mathbf{B} = [\mathrm{Re}\{\mathbf{b}\}^{\mathrm{T}}, \mathrm{Im}\{\mathbf{b}\}^{\mathrm{T}}]^{\mathrm{T}}[\mathrm{Re}\{\mathbf{b}\}^{\mathrm{T}}, \mathrm{Im}\{\mathbf{b}\}^{\mathrm{T}}]$, where $\mathrm{Im}\{\cdot\}$ takes the imaginary part. Given normalizing and power scaling factors, CPU frequencies, and data allocation, the subproblem regarding receive beamformers can be transformed to the following semidefinite programming problem:

$$\begin{aligned}
\min_{\{\mathbf{V}_k\},\mathbf{B}} \quad & \sum\nolimits_{k=1}^{K} \frac{C_{9,k}}{\mathrm{tr}(\mathbf{V}_k\mathbf{H}_k^{\mathrm{D}})} + \sum\nolimits_{k=1}^{K} \frac{C_{10}}{\mathrm{tr}(\mathbf{B}\mathbf{H}_k^{\mathrm{G}})} && \text{(41a)}\\
\text{s.t.} \quad & -p_{\max}\mathrm{tr}(\mathbf{V}_k\mathbf{H}_k^{\mathrm{D}}) + \zeta_k \le 0, \forall k \in \mathcal{K}, && \text{(41b)}\\
& -p_{\max}\mathrm{tr}(\mathbf{B}\mathbf{H}_k^{\mathrm{G}}) + \omega \le 0, \forall k \in \mathcal{K}, && \text{(41c)}\\
& \mathrm{tr}(\mathbf{V}_k) = 1, \forall k \in \mathcal{K}, && \text{(41d)}\\
& \mathrm{tr}(\mathbf{B}) = 1, && \text{(41e)}\\
& \mathbf{V}_k \succeq \mathbf{0}, \forall k \in \mathcal{K}, && \text{(41f)}\\
& \mathbf{B} \succeq \mathbf{0}, && \text{(41g)}\\
& \mathrm{rank}(\mathbf{V}_k) = 1, \forall k \in \mathcal{K}, && \text{(41h)}\\
& \mathrm{rank}(\mathbf{B}) = 1, && \text{(41i)}
\end{aligned}$$

where $C_{9,k} = D\bar{C}\theta_k\zeta_k/[B\log(1+\zeta_k/\sigma^2)], \forall k \in \mathcal{K}$, $C_{10} = \lceil Q/M \rceil T_s\omega$, and $\mathrm{tr}(\cdot)$ denotes the trace. Moreover, matrices $\mathbf{H}_k^{\mathrm{D}}$ and $\mathbf{H}_k^{\mathrm{G}}$ are defined in (42) and (43), respectively. However, problem (41) is non-convex due to the rank-one constraints (41h) and (41i).

$$\mathbf{H}_k^{\mathrm{D}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} + \mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} & \mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} - \mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} \\ \mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} - \mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} & \mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} + \mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{D}}\}^{\mathrm{T}} \end{bmatrix}, \forall k \in \mathcal{K}, \tag{42}$$

$$\mathbf{H}_k^{\mathrm{G}} = \begin{bmatrix} \mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} + \mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} & \mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} - \mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} \\ \mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} - \mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} & \mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Re}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} + \mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}\mathrm{Im}\{\mathbf{h}_k^{\mathrm{G}}\}^{\mathrm{T}} \end{bmatrix}, \forall k \in \mathcal{K}. \tag{43}$$

We employ a difference-of-convex-functions (DC) programming method to address the non-convexity of the rank-one constraints [35]. Specifically, constraints (41h) and (41i) are equivalent to (44) and (45), respectively, given by

$$\mathrm{tr}(\mathbf{V}_k) - \|\mathbf{V}_k\|_2 = 0, \forall k \in \mathcal{K}, \tag{44}$$

$$\mathrm{tr}(\mathbf{B}) - \|\mathbf{B}\|_2 = 0, \tag{45}$$

where $\|\cdot\|_2$ denotes the matrix 2-norm. Both (44) and (45) are still non-convex due to $-\|\mathbf{V}_k\|_2$ and $-\|\mathbf{B}\|_2$. We substitute $\|\mathbf{V}_k\|_2$ and $\|\mathbf{B}\|_2$ with their linearizations $\|\mathbf{V}_k^{(n_1)}\|_2 + \mathrm{tr}((\mathbf{V}_k - \mathbf{V}_k^{(n_1)})^{\mathrm{T}}\dot{\mathbf{V}}_k^{(n_1)})$ and $\|\mathbf{B}^{(n_1)}\|_2 + \mathrm{tr}((\mathbf{B} - \mathbf{B}^{(n_1)})^{\mathrm{T}}\dot{\mathbf{B}}^{(n_1)})$, respectively. Here, $\mathbf{V}_k^{(n_1)}$ and $\mathbf{B}^{(n_1)}$ are obtained in the n_1-th DC iteration, while $\dot{\mathbf{V}}_k^{(n_1)}$ and $\dot{\mathbf{B}}^{(n_1)}$ denote the subgradients obtained in the n_1-th DC iteration. Then, by substituting (41h) and (41i) with their linearizations, we add them to objective (41a) as regularizers to convexify problem (41) as follows:

$$\begin{aligned} \min_{\{\mathbf{V}_k\},\mathbf{B}} \quad & \sum\nolimits_{k=1}^{K} \frac{C_{9,k}}{\mathrm{tr}(\mathbf{V}_k\mathbf{H}_k^{\mathrm{D}})} + \sum\nolimits_{k=1}^{K} \frac{C_{10}}{\mathrm{tr}(\mathbf{B}\mathbf{H}_k^{\mathrm{G}})} \\ & + \sum\nolimits_{k=1}^{K} \beta[\mathrm{tr}(\mathbf{V}_k) - \mathrm{tr}(\mathbf{V}_k^{\mathrm{T}}\dot{\mathbf{V}}_k^{(n_1)})] \\ & + \beta[\mathrm{tr}(\mathbf{B}) - \mathrm{tr}(\mathbf{B}^{\mathrm{T}}\dot{\mathbf{B}}^{(n_1)})] & \text{(46a)} \\ \text{s.t.} \quad & \text{(41b) – (41g)}, & \text{(46b)} \end{aligned}$$

where $\beta > 0$ denotes the penalty factor. In view of the convexity of problem (46), it can be solved using standard optimization toolkits like CVX [36]. The DC-based algorithm for solving problem (41) is summarized in Algorithm 1.

Algorithm 1 A DC-Based Algorithm for Optimizing Receive Beamformers

```
Input: Feasible receive beamformers $(\{\mathbf{v}_k^{(0)}\}, \mathbf{b}^{(0)})$, the maximum numbers of iterations $N_1$, and $n_1 = 0$.
Calculate $\{\mathbf{V}_k^{(0)}\}$ and $\mathbf{B}^{(0)}$.
repeat
    Update $n_1 \leftarrow n_1 + 1$.
    Calculate subgradients $\{\dot{\mathbf{V}}_k^{(n_1-1)}\}$ and $\dot{\mathbf{B}}^{(n_1-1)}$ using $\{\mathbf{V}_k^{(n_1-1)}\}$ and $\mathbf{B}^{(n_1-1)}$, respectively.
    Obtain $\{\mathbf{V}_k^{(n_1)}\}$ and $\mathbf{B}^{(n_1)}$ by solving problem (46).
until The objective (41a) converges or $n_1 \geq N_1$.
Recover $\{\mathbf{v}_k^*\}$ and $\mathbf{b}^*$ based on $\{\mathbf{V}_k^{(n_1)}\}$ and $\mathbf{B}^{(n_1)}$, respectively.
Output: Optimized receive beamformers $\{\mathbf{v}_k^*\}$ and $\mathbf{b}^*$.
```

3) CPU Frequencies: Given normalizing and power scaling factors, receive beamformers, and data allocation, by introducing an auxiliary variable $\tau_2 \geq T^{\mathrm{ALL}}$, the subproblem regarding CPU frequencies are given by

$$\begin{aligned} \min_{\{\hat{f}_k\},\tilde{f},\tau_2} \quad & \sum\nolimits_{k=1}^{K} C_{11,k}\hat{f}_k^2 + C_{12}\tilde{f}^2 & \text{(47a)} \\ \text{s.t.} \quad & \tau_2 - T_{\max} \leq 0, & \text{(47b)} \\ & \frac{C_{13,k}}{\hat{f}_k} - \tau_2 + T^{\mathrm{G}} \leq 0, \forall k \in \mathcal{K}, & \text{(47c)} \\ & \frac{C_{14}}{\tilde{f}} - \tau_2 + \max_{k \in \mathcal{K}}\{T_k^{\mathrm{D}}\} \leq 0, & \text{(47d)} \\ & \text{(32h) and (32i)}, & \text{(47e)} \end{aligned}$$

where $C_{11,k} = C_{13,k}\hat{\kappa}, \forall k \in \mathcal{K}$, $C_{12} = C_{14}\tilde{\kappa}$, $C_{13,k} = D(1 - \theta_k)\hat{C}_k, \forall k \in \mathcal{K}$, and $C_{14} = D\tilde{C}\sum_{k=1}^{K}\theta_k$. Problem (47) is jointly convex with respect to $\{\hat{f}_k\}$, $\tilde{f}$, and τ_2. Hence, the closed-form solutions are derived by solving its Karush-Kuhn-Tucker (KKT) conditions, as presented in Lemma 1.

Lemma 1. *By solving the KKT conditions, the closed-form solution to problem (47) is given by*

$$\hat{f}_k^* = \frac{C_{13,k}}{T_{\max} - T^{\mathrm{G}}}, \forall k \in \mathcal{K}, \tag{48}$$

$$\tilde{f}^* = \frac{C_{14}}{T_{\max} - \max_{k \in \mathcal{K}}\{T_k^{\mathrm{D}}\}}, \tag{49}$$

$$\tau_2^* = T_{\max}. \tag{50}$$

Proof: Please refer to Appendix D in [33]. □

4) Data Allocation: Given normalizing and power scaling factors, receive beamformers, and CPU frequencies, by introducing an auxiliary variable $\tau_3 \geq T^{\mathrm{ALL}}$, the subproblem regarding data allocation is reduced to

$$\begin{aligned} \min_{\{\theta_k\},\tau_3} \quad & \sum\nolimits_{k=1}^{K} C_{15,k}\theta_k & \text{(51a)} \\ \text{s.t.} \quad & \tau_3 - T_{\max} \leq 0, & \text{(51b)} \\ & C_{16,k}\theta_k + C_{17}\sum\nolimits_{k'=1}^{K}\theta_{k'} - \tau_3 \leq 0, \forall k \in \mathcal{K}, & \text{(51c)} \\ & -C_{18,k}\theta_k - \tau_3 + T^{\mathrm{G}} + C_{18,k} \leq 0, \forall k \in \mathcal{K}, & \text{(51d)} \\ & 0 \leq \theta_k \leq C_{19}, \forall k \in \mathcal{K}, & \text{(51e)} \end{aligned}$$

where $C_{15,k} = \zeta_k C_{16,k}/|\mathbf{v}_k^{\mathrm{H}}\mathbf{h}_k^{\mathrm{D}}|^2 - D\hat{C}_k\hat{\kappa}\hat{f}_k^2 + D\tilde{C}\tilde{\kappa}\tilde{f}^2, \forall k \in \mathcal{K}$, $C_{16,k} = D\bar{C}/[B\log(1 + \zeta_k/\sigma^2)], \forall k \in \mathcal{K}$, $C_{17} = \tilde{C}D/\tilde{f}$, $C_{18,k} = \hat{C}_k D/\hat{f}_k, \forall k \in \mathcal{K}$, and $C_{19} = \theta_{\max}$. Problem (51) is a linear programming problem regarding $\{\theta_k\}$ and τ_3, which can be effectively solved using CVX.

5) Overall Algorithm for Non-Stable Region: The overall algorithm proposed for solving problem (32) in the non-stable region $\mathcal{R}^{\mathrm{NS}}$ is summarized in Algorithm 2, where a variable with the superscript (n_2) denotes its value obtained at the n_2-th iteration. The solutions to subproblems (34), (41), (47), and (51) create a series of non-increasing objective values for (32a). Meanwhile, as the objective (32a) essentially represents energy consumption, it is naturally lower

Algorithm 2 Proposed Algorithm for Solving Problem (32)

```
Input: A feasible solution $(\{\zeta_k^{(0)}\}, \nu^{(0)}, \omega^{(0)}, \mathbf{b}^{(0)}, \{\mathbf{v}_k^{(0)}\}, \{\hat{f}_k^{(0)}\}, \tilde{f}^{(0)}, \{\theta_k^{(0)}\})$, the maximum numbers of iterations $N_2$, and $n_2 = 0$.
repeat
    Update $n_2 \leftarrow n_2 + 1$.
    Given $\mathbf{b}^{(n_2-1)}$, $\{\mathbf{v}_k^{(n_2-1)}\}$, $\{\hat{f}_k^{(n_2-1)}\}$, $\tilde{f}^{(n_2-1)}$, $\{\theta_k^{(n_2-1)}\}$, calculate $\nu^{(n_2)}$, $\omega^{(n_2)}$, and $\{\zeta_k^{(n_2)}\}$ via (36), (37), and (40), respectively.
    Given $\{\zeta_k^{(n_2)}\}$, $\nu^{(n_2)}$, $\omega^{(n_2)}$, $\{\hat{f}_k^{(n_2-1)}\}$, $\tilde{f}^{(n_2-1)}$, $\{\theta_k^{(n_2-1)}\}$, obtain $\mathbf{b}^{(n_2)}$ and $\{\mathbf{v}_k^{(n_2)}\}$ by using Algorithm 1.
    Given $\{\zeta_k^{(n_2)}\}$, $\nu^{(n_2)}$, $\omega^{(n_2)}$, $\mathbf{b}^{(n_2)}$, $\{\mathbf{v}_k^{(n_2)}\}$, $\{\theta_k^{(n_2-1)}\}$, calculate $\{\hat{f}_k^{(n_2)}\}$ and $\tilde{f}^{(n_2)}$ via (48) and (49), respectively.
    Given $\{\zeta_k^{(n_2)}\}$, $\nu^{(n_2)}$, $\omega^{(n_2)}$, $\mathbf{b}^{(n_2)}$, $\{\mathbf{v}_k^{(n_2)}\}$, $\{\hat{f}_k^{(n_2)}\}$, $\tilde{f}^{(n_2)}$, obtain $\{\theta_k^{(n_2)}\}$ by solving problem (51).
until The objective (32a) converges or $n_2 \geq N_2$.
Output: The optimized solution $(\{\zeta_k^{(n_2)}\}, \nu^{(n_2)}, \omega^{(n_2)}, \mathbf{b}^{(n_2)}, \{\mathbf{v}_k^{(n_2)}\}, \{\hat{f}_k^{(n_2)}\}, \tilde{f}^{(n_2)}, \{\theta_k^{(n_2)}\})$.
```

bounded by zero. Therefore, the convergence of Algorithm 2 is ensured. The main complexity of Algorithm 2 is given by $\mathcal{O}((K+2)N_2+(K+1)N_r^{4.5}N_1N_2+(k+1)N_2+(K+1)^3N_2)$. Specifically, when invoking CVX to solve a convex problem, the standard interior-point method is adopted. Thus, the complexity of optimizing the normalizing and power scaling factors, i.e., ν_t, ω, and $\{\zeta_k\}$, is $\mathcal{O}(K+2)$. The complexity of performing Algorithm 1 is $\mathcal{O}((K+1)N_r^{4.5}N_1)$ [37]. The complexities of optimizing CPU frequencies, i.e., $\{\hat{f}_k\}$ and $\tilde{f}$, and data allocation, i.e., $\{\theta_k\}$, can be given by $\mathcal{O}(K+1)$ and $\mathcal{O}((K+1)^3)$ [38], respectively.

B. Algorithm for Stable Region

We address problem (33) in the stable region $\mathcal{R}^{\rm S}$ by decoupling it into four subproblems as well. The specific subproblems and their solutions are presented as follows:

1) Normalizing and Power Scaling Factors: Given receive beamformers, CPU frequencies, and data allocation, by introducing an auxiliary variable $\tau_4 \geq T^{\rm ALL}$, the subproblem regarding normalizing and power scaling factors is reduced to

$$\begin{aligned}
\min_{\{\zeta_k\},\nu,\omega,\tau_4} \quad & \sum_{k=1}^{K}\left[C_{1,k}\frac{\zeta_k}{\log(1+\frac{\zeta_k}{\sigma^2})}+C_{2,k}\omega\right] && \text{(52a)}\\
\text{s.t.} \quad & \tau_4 - T_{\max} \leq 0, && \text{(52b)}\\
& \frac{C_{3,k}}{\log(1+\frac{\zeta_k}{\sigma^2})}+T^{\rm E}-\tau_4 \leq 0, \forall k \in \mathcal{K}, && \text{(52c)}\\
& \max_{k\in\mathcal{K}}\{T_k^{\rm L}\}+T^{\rm G}-\tau_4 \leq 0, && \text{(52d)}\\
& C_{20}\nu + C_{21} \leq 0, && \text{(52e)}\\
& C_{22}\nu - 2\sqrt{\omega}\sqrt{\nu}+\omega+C_8 \leq 0, && \text{(52f)}\\
& \sqrt{\omega}=\sqrt{\nu}, && \text{(52g)}\\
& \text{(34e), and (34f)}, && \text{(52h)}
\end{aligned}$$

where $C_{20} = A^2 - \epsilon_3\mu(4\mu - L)/L$, $C_{21} = Q\sigma^2/2$, and $C_{22} = 1 - \epsilon_4 K$. Problem (52) is non-convex because of the non-convex objective (52a) and constraint (52f). We further decouple it into two subproblems: one involving ν and ω, and the other involving $\{\zeta_k\}$ and τ_4.

By plugging in constraint (52g), i.e., $\sqrt{\omega} = \sqrt{\nu}$, the subproblem regarding ν and ω degrades to a univariate linear programming problem of ν, given by

$$\begin{aligned}
\min_{\nu} \quad & \left(\sum_{k=1}^{K} C_{2,k}\right)\nu && \text{(53a)}\\
\text{s.t.} \quad & \max\left\{-\frac{C_{21}}{C_{20}}, \frac{C_8}{1-C_{22}}\right\} \leq \nu \leq C_5. && \text{(53b)}
\end{aligned}$$

Since $C_{2,k} \geq 0, \forall k \in \mathcal{K}$, the objective (53) monotonously increases with ν. Thus, the optimal ν and ω are given by

$$\nu^* = \omega^* = \max\left\{-\frac{C_{21}}{C_{20}}, \frac{C_8}{1-C_{22}}\right\}. \tag{54}$$

On the other hand, it can be verified that the subproblem involving $\{\zeta_k\}$ and τ_4 has a form identical to problem (38). Hence, due to the problem similarity, the optimal $\{\zeta_k\}$ can be given by (40), while the optimal τ_4 can be determined by

$$\tau_4^* = T_{\max}. \tag{55}$$

2) Receive Beamformers: Given normalizing and power scaling factors, CPU frequencies, and data allocation, by using the matrices $\{\mathbf{V}_k\}$ and $\mathbf{B}$ defined in Section IV-A2, one can find that the subproblem of receive beamformers in the stable region $\mathcal{R}^{\rm S}$ has the same form as problem (41). Due to the problem similarity, one can obtain the optimal receiver beamformers $\{\mathbf{v}_k^*\}$ and $\mathbf{b}^*$ by applying Algorithm 1.

3) CPU Frequencies: Given normalizing and power scaling factors, receive beamformers, and data allocation, the subproblem of CPU frequencies $\{\hat{f}_k\}$ and $\tilde{f}$ is the same as problem (47), which is convex. Based on Lemma 1, the optimal CPU frequencies $\{\hat{f}_k^*\}$ and $\tilde{f}^*$ in the stable region $\mathcal{R}^{\rm S}$ can be obtained using (48) and (49), respectively.

4) Data Allocation: Given normalizing and power scaling factors, receive beamformers, and CPU frequencies, by introducing an auxiliary variable $\tau_5 \geq T^{\rm ALL}$, the subproblem of data allocation is reduced to

$$\begin{aligned}
\min_{\{\theta_k\},\tau_5} \quad & \sum_{k=1}^{K} C_{15,k}\theta_k && \text{(56a)}\\
\text{s.t.} \quad & \tau_5 - T_{\max} \leq 0, && \text{(56b)}\\
& C_{16,k}\theta_k + C_{17}\sum_{k'=1}^{K}\theta_{k'} - \tau_5 \leq 0, \forall k \in \mathcal{K}, && \text{(56c)}\\
& -C_{18,k}\theta_k - \tau_5 + T^{\rm G} + C_{18,k} \leq 0, \forall k \in \mathcal{K}, && \text{(56d)}\\
& C_{23} \leq \theta_k \leq 1, \forall k \in \mathcal{K}, && \text{(56e)}
\end{aligned}$$

where $C_{23} = \theta_{\min}$. Problem (56) is a linear programming problem, which can be effectively solve using CVX.

5) Overall Algorithm for Stable Region: The overall algorithm for solving problem (33) in the stable region $\mathcal{R}^{\rm S}$ is summarized in Algorithm 3. In Algorithm 3, a variable with the superscript (n_3) refers to its value obtained in the n_3-th iteration. The convergence of Algorithm 3 can be analyzed similarly to that of Algorithm 2. In addition, the complexity

Algorithm 3 Proposed Algorithm for Solving Problem (33)

```
Input: A feasible solution ($\{\zeta_k^{(0)}\}$, $\nu^{(0)}$, $\omega^{(0)}$, $\mathbf{b}^{(0)}$, $\{\mathbf{v}_k^{(0)}\}$, $\{\hat{f}_k^{(0)}\}$, $\tilde{f}^{(0)}$, $\{\theta_k^{(0)}\}$), the maximum numbers of iterations $N_3$, and $n_3 = 0$.
repeat
    Update $n_3 \leftarrow n_3 + 1$.
    Given $\mathbf{b}^{(n_3-1)}$, $\{\mathbf{v}_k^{(n_3-1)}\}$, $\{\hat{f}_k^{(n_3-1)}\}$, $\tilde{f}^{(n_3-1)}$, $\{\theta_k^{(n_3-1)}\}$, calculate $\nu^{(n_3)}$ and $\omega^{(n_3)}$ via (54), and calculate $\{\zeta_k^{(n_3)}\}$ via (40).
    Given $\{\zeta_k^{(n_3)}\}$, $\nu^{(n_3)}$, $\omega^{(n_3)}$, $\{\hat{f}_k^{(n_3-1)}\}$, $\tilde{f}^{(n_3-1)}$, $\{\theta_k^{(n_3-1)}\}$, obtain $\mathbf{b}^{(n_3)}$ and $\{\mathbf{v}_k^{(n_3)}\}$ by using Algorithm 1.
    Given $\{\zeta_k^{(n_3)}\}$, $\nu^{(n_3)}$, $\omega^{(n_3)}$, $\mathbf{b}^{(n_3)}$, $\{\mathbf{v}_k^{(n_3)}\}$, $\{\theta_k^{(n_3-1)}\}$, calculate $\{\hat{f}_k^{(n_3)}\}$ and $\tilde{f}^{(n_3)}$ via (48) and (49), respectively.
    Given $\{\zeta_k^{(n_3)}\}$, $\nu^{(n_3)}$, $\omega^{(n_3)}$, $\mathbf{b}^{(n_3)}$, $\{\mathbf{v}_k^{(n_3)}\}$, $\{\hat{f}_k^{(n_3)}\}$, $\tilde{f}^{(n_3)}$, obtain $\{\theta_k^{(n_3)}\}$ by solving problem (56).
until The objective (33a) converges or $n_3 \geq N_3$.
Output: The optimized solution ($\{\zeta_k^{(n_3)}\}$, $\nu^{(n_3)}$, $\omega^{(n_3)}$, $\mathbf{b}^{(n_3)}$, $\{\mathbf{v}_k^{(n_3)}\}$, $\{\hat{f}_k^{(n_3)}\}$, $\tilde{f}^{(n_3)}$, $\{\theta_k^{(n_3)}\}$).
```

TABLE I
SIMULATION PARAMETERS

Parameters	**Value**
Maximum transmit power	$\hat{p}_{\max} = 23$ dBm
Bandwidth for data uploading	$B = 10$ kHz
Noise power	$\sigma^2 = -80$ dBm
AirComp symbol duration	$T_s = 1$ ms
Local dataset size	$D = 3 \times 10^3$
Penalty factor	$\beta = 1$
Number of symbols in an AirComp symbol	$M = 14$
Maximum CPU frequencies of the BS and devices	$\tilde{f}_{\max} = 10$ GHz, $\hat{f}_{\max} = 1$ GHz
Number of CPU circles for processing a data sample	$\tilde{C} = 1 \times 10^8$, $\hat{C}_k \in [1.5, 2.8] \times 10^8$
Effective switched capacitance of the BS and devices	$\tilde{\kappa} = 1 \times 10^{-28}$, $\hat{\kappa} = 1 \times 10^{-28}$
Maximum and minimum ratios of SL data	$\theta_{\max} = 0.3$, $\theta_{\min} = 0.2$

of Algorithm 3 can be given by $\mathcal{O}((K+2)N_3 + (K+1)N_r^{4.5}N_1N_3 + (k+1)N_3 + (K+1)^3N_3)$.

V. SIMULATION RESULTS

A. Simulation Setup

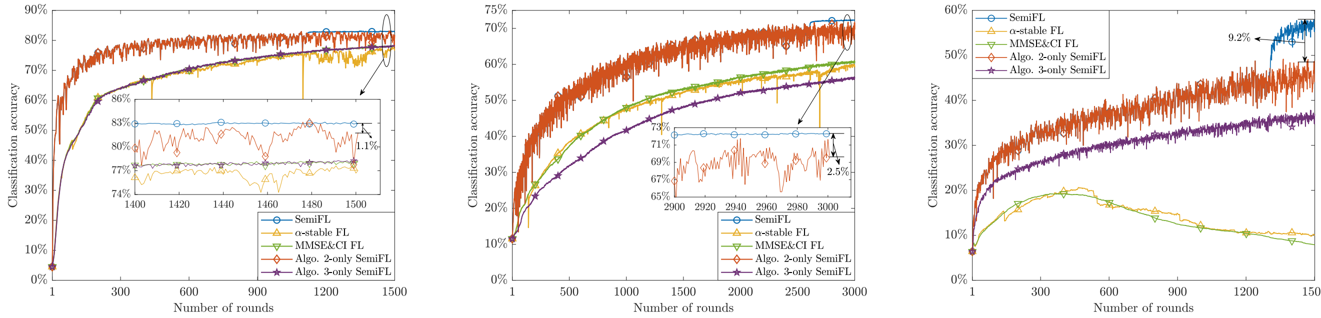
We build a virtual urban area using MATLAB, where $K = 20$ single-antenna devices are randomly distributed in a square area of 100 m × 100 m. A BS equipped with $N_r = 16$ antennas is located at the center. We adopt a clustered delay line multiple-input multiple-output link-level fading channel model to generate the channel coefficient vectors for gradient aggregation and data uploading [39], i.e., $\{\mathbf{h}_k^{\mathrm{G}}\}$ and $\{\mathbf{h}_k^{\mathrm{D}}\}$. To verify learning performance, we train a MLP, a CNN, and a ResNet to classify the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, respectively. The MLP has 3 fully-connected hidden layers, and the learning rate for training the MLP is set to $\eta = 0.01$. The CNN contains 3 convolutional layers, 3 max-pooling layers, and 2 fully-connected layers, while the learning rate is set to $\eta = 0.001$. The ResNet has 11 convolution layers and a fully-connected layer. The learning rate for training the ResNet is set to $\eta = 0.001$. Unless otherwise specified, other main simulation parameters are listed in Table I. Additional simulation results in terms of energy consumption can be find in the Appendix A of [33].

To validate the effectiveness of over-the-air distortion in accelerating convergence of SemiFL, we consider the following state-of-the-art benchmarks for comparison:

- **FL with noise yielding α-stable distribution (α-stable FL)** [23]: Devices upload only local gradients to the BS for aggregation, where the aggregation noise obeys an α-stable distribution. This scheme is reported to improve model generalization.
- **FL with MMSE and CI (MMSE&CI FL)** [40], [41]: All devices upload only local gradients to the BS for aggregation, where the variables are optimized using the CI&MMSE scheme to suppress over-the-air distortion.
- **SemiFL with only Algorithm 2 (Algo. 2-only SemiFL)**: Only Algorithm 2 is applied in both the non-stable region $\mathcal{R}^{\mathrm{NS}}$ and the stable region $\mathcal{R}^{\mathrm{S}}$.
- **SemiFL with only Algorithm 3 (Algo. 3-only SemiFL)**: Only Algorithm 3 is applied in both the non-stable region $\mathcal{R}^{\mathrm{NS}}$ and the stable region $\mathcal{R}^{\mathrm{S}}$.

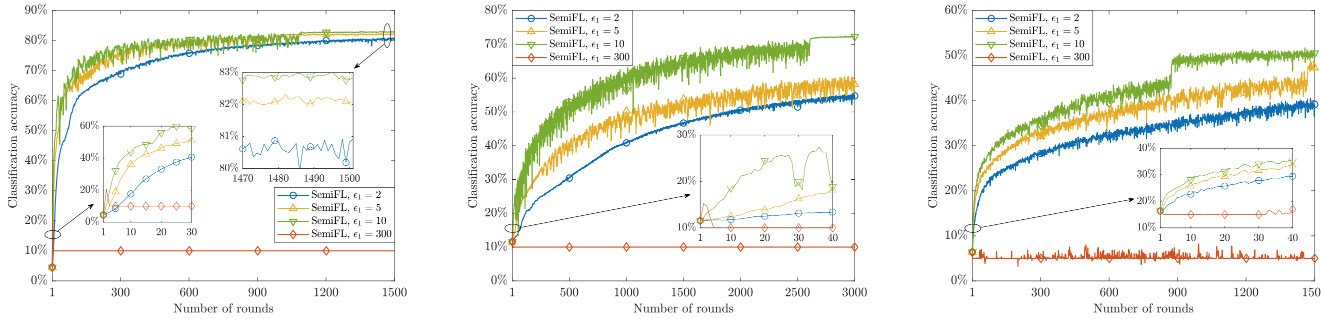
Fig. 3 shows the learning performance comparisons between the proposed SemiFL and state-of-the-art benchmarks on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets. In Figs. 3(a), 3(b), and 3(c), it is seen that the convergence of SemiFL is significantly accelerated by leveraging over-the-air distortion, achieving faster convergence than benchmarks. Upon reaching the stable region, SemiFL exhibits stable and improved final convergence by invoking Algorithm 3, obtaining accuracy gains of 1.1%, 2.5%, and 9.2% on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, respectively, compared to Algo. 2-only SemiFL. Meanwhile, it is also seen that using either Algorithm 2 or Algorithm 3 alone results in inferior convergence than the combined usage of them, verifying the effectiveness of the two-region MSE threshold configuration outlined in Remark 3. Additionally, it is observed that SemiFL with proposed algorithms outperforms the widely adopted MMSE&CI FL scheme in all cases. This confirms that the conventional distortion-suppressing criterion for AirComp-based gradient aggregation is overly conservative. Moreover, Fig. 3(c) shows that both the MMSE&CI FL and α-stable FL fail to train the complex ResNet, whereas SemiFL with proposed algorithms keeps admirable learning performance.

Fig. 4 shows the impact of different ϵ_1 values on the learning performance of SemiFL. In Figs. 4(a), 4(b), and 4(c), as ϵ_1 increases from 2 to 10, we see that the convergence acceleration effect of over-the-air distortion is enhanced in early rounds, while higher final accuracy is obtained in the end. This confirms that ϵ_1, or equivalently the power scaling factor-to-normalizing factor ratio $\sqrt{\omega}/\sqrt{\nu}$, is a key factor in modulating



(a) Training MLP on the Fashion-MNIST dataset. (b) Training CNN on the CIFAR-10 dataset. (c) Training ResNet on the CIFAR-100 dataset.

Fig. 3. Learning performance comparison between SemiFL and benchmarks on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, where $\epsilon_1 = 10$, $\epsilon_2 = 1$, $\epsilon_3 = 0.8$, and $\epsilon_4 = 0.01$.



(a) Training MLP on the Fashion-MNIST dataset. (b) Training CNN on the CIFAR-10 dataset. (c) Training ResNet on the CIFAR-100 dataset.

Fig. 4. Learning performance comparison of the proposed SemiFL on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets with different ϵ_1 values, where $\epsilon_3 = 0.8$ and $\epsilon_4 = 0.01$. Note that we set $\epsilon_2 = 1$ when $\epsilon_1 = 2$ or 5, and set $\epsilon_2 = 5$ when $\epsilon_1 = 10$. When $\epsilon_1 = 300$, a sufficiently large ϵ_2 is adopted.

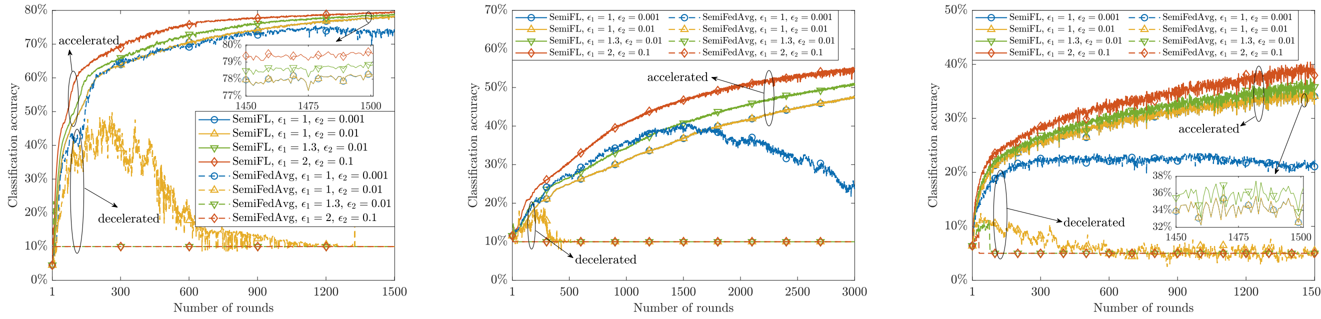
the acceleration effect of over-the-air distortion. However, Figs. 4(a), 4(b), and 4(c) also show that an excessively large ϵ_1 leads to training failure for SemiFL, as excessive over-the-air distortion collapses the global model. This indicates that the value of ϵ_1 should be carefully moderated.

Fig. 5 shows the learning performance comparisons between SemiFL and a benchmark, named SemiFL with federated averaging (SemiFedAvg). The only difference of SemiFedAvg is that devices upload model parameters, i.e., weights and biases, to the BS for aggregation, rather than gradients. In Figs. 5(a), 5(b), and 5(c), it is intriguing to see that as ϵ_1 and ϵ_2 increase, the convergence of SemiFL is gradually accelerated, whereas the convergence of SemiFedAvg is significantly decelerated. This is because over-the-air distortion is directly imposed on model parameters in SemiFedAvg, fundamentally disrupting the global model. For the proposed SemiFL, over-the-air distortion only affects uploaded gradients, leaving the global model intact. This also highlights a key insight: the convergence acceleration effect of over-the-air distortion is tailored to AirComp-based gradient aggregation. Moreover, Figs. 5(a), 5(b), and 5(c) show that increasing ϵ_2 while keeping ϵ_1 constant cannot trigger the convergence acceleration effect of over-the-air distortion, as evidenced by the overlap of the yellow and blue solid lines. However, it should be emphasized that a large ϵ_2 enables the usage of a large ϵ_1, eventually enhancing the convergence acceleration effect of over-the-air distortion.

Figs. 6 and 7 compare the learning performance and energy consumption between SemiFL and CL, respectively. In CL, the BS collects all raw data from devices in the first round, and utilizes the collected data to train the global model round by round. In Figs. 6(a), 6(b), and 6(c), the proposed SemiFL uniformly outperforms CL in all cases by achieving accelerated convergence and final accuracy, particularly with gains of 4.47%, 18.39%, and 7.89% on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, respectively. Moreover, in Fig. 7, we see that SemiFL dramatically reduces energy consumption compared to CL, saving 98.36%, 98.46%, and 97.82% energy for training MLP, CNN, and ResNet, respectively. This is because SemiFL replaces energy-consuming entire dataset uploading with partial intermediate outputs uploading. The above observations validate that SemiFL with proposed algorithms offers a more energy-efficient learning framework than conventional CL which naively gathers all data at the BS.

VI. Conclusion

In this paper, we proposed a novel approach that harnesses over-the-air distortion to accelerate the convergence of SemiFL. The convergence acceleration effect of over-the-air distortion was triggered by intentionally permitting a high MSE threshold in the non-stable region, so as to amplify the ratio of the power scaling factor to the normalizing factor. To preserve data privacy, the new SemiFL framework incorporated FL with SL, whereby the global model was separated into shallow and deep layers. Devices processed SL data through the shallow layers and then sent the resulting

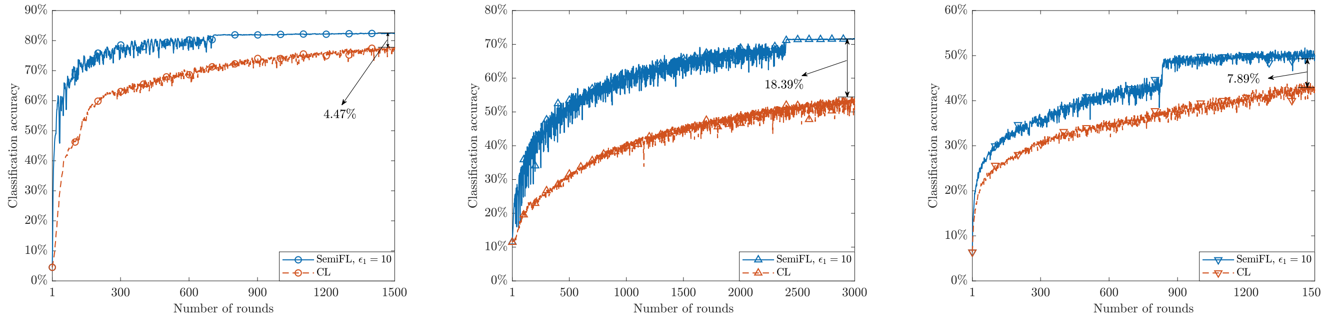


(a) Training MLP on the Fashion-MNIST dataset.

(b) Training CNN on the CIFAR-10 dataset.

(c) Training ResNet on the CIFAR-100 dataset.

Fig. 5. Learning performance comparison between the proposed SemiFL and SemiFedAvg on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets with different ϵ_1 and ϵ_2 values, where $\epsilon_3 = 0.8$ and $\epsilon_4 = 0.01$.



(a) Training MLP on the Fashion-MNIST dataset.

(b) Training CNN on the CIFAR-10 dataset.

(c) Training ResNet on the CIFAR-100 dataset.

Fig. 6. Learning performance comparison between the proposed SemiFL and CL on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets, where $\epsilon_1 = 10$, $\epsilon_2 = 1$, $\epsilon_3 = 0.8$, and $\epsilon_4 = 0.01$. Note that $T_{\max}$ is set to 500 s for MLP and CNN, and 650 s for ResNet.

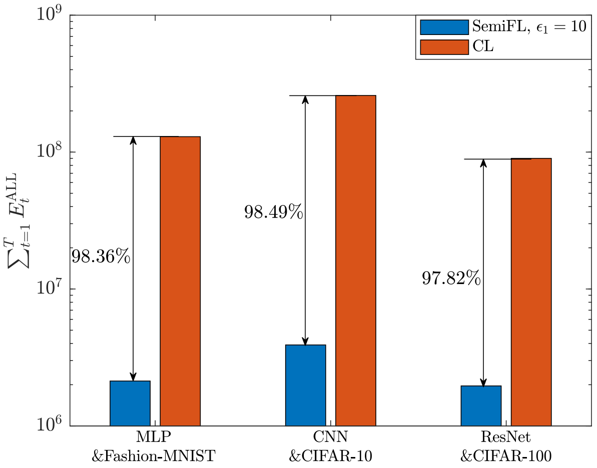


Fig. 7. Energy consumption comparison between the proposed SemiFL and CL on the Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets.

intermediate outputs to the BS for training the deep layers, thereby prevent the BS from accessing raw data. We derived a lower bound for the expected loss reduction between two consecutive rounds in the non-stable region, revealing that high distortion helps accelerate convergence by boosting loss descent. This theoretically substantiated the convergence acceleration effect of over-the-air distortion. However, another upper bound for the optimality gap indicated that suppressing over-the-air distortion was indispensable for enhancing the final convergence. In view of the antagonistic roles of over-the-air distortion in different regions, we formulated two distinct energy consumption minimization problems—one for each type of region—based on a proposed two-region MSE threshold configuration scheme. We developed two resource allocation algorithms to address the two problems, where closed-form solutions were provided. Simulation results using three AI models and three real datasets confirmed the effectiveness of the proposed approach and algorithms in accelerating convergence and reducing energy consumption.

References

[1] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji *et al.*, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, May 2020.

[2] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li *et al.*, "Federated learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1622–1658, 3rd Quart. 2021.

[3] M. Le, T. Huynh-The, T. Do-Duy, T.-H. Vu, W.-J. Hwang *et al.*, "Applications of distributed machine learning for the internet-of-things: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, 2024, early access, doi: 10.1109/COMST.2024.3427324.

[4] X. Liu, Y. Deng, A. Nallanathan, and M. Bennis, "Federated learning and meta learning: Approaches, applications, and directions," *IEEE Commun. Surv. Tutorials*, vol. 26, no. 1, pp. 571–618, 1st Quart. 2024.

[5] Q. Duan, J. Huang, S. Hu, R. Deng, Z. Lu *et al.*, "Combining federated learning and edge computing toward ubiquitous intelligence in 6G network: Challenges, recent advances, and future directions," *IEEE*

Commun. Surv. Tutorials, vol. 25, no. 4, pp. 2892–2950, 4th, Quart. 2023.

[6] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar *et al.*, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020.

[7] W. Ni, J. Zheng, and H. Tian, "Semi-federated learning for collaborative intelligence in massive IoT networks," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 942–11 943, 2023.

[8] A. M. Elbir, S. Coleri, A. K. Papazafeiropoulos, P. Kourtessis, and S. Chatzinotas, "A hybrid architecture for federated and centralized learning," *IEEE Trans. Cognit. Commun. Networking*, vol. 8, no. 3, pp. 1529–1542, Sep. 2022.

[9] J. Zheng, W. Ni, H. Tian, D. Gündüz, T. Q. S. Quek *et al.*, "Semi-federated learning: Convergence analysis and optimization of a hybrid learning framework," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 9438–9456, Dec. 2023.

[10] C. Feng, H. H. Yang, S. Wang, Z. Zhao, and T. Q. S. Quek, "Hybrid learning: When centralized learning meets federated learning in the mobile edge computing systems," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 7008–7022, Dec. 2023.

[11] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis *et al.*, "Mix2FLD: Downlink federated learning after uplink federated distillation with two-way mixup," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2211–2215, Oct. 2020.

[12] Q. Zhang, L. T. Yang, and Z. Chen, "Privacy preserving deep computation model on cloud for big data feature learning," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1351–1362, May 2016.

[13] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 3, pp. 1877–1908, 3rd Quart. 2023.

[14] J. Zhu, Y. Shi, Y. Zhou, C. Jiang, W. Chen *et al.*, "Over-the-air federated learning and optimization," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 16 996–17 020, May 2024.

[15] G. Zhu and K. Huang, "MIMO over-the-air computation for high-mobility multimodal sensing," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6089–6103, Aug. 2019.

[16] L. Li, C. Huang, D. Shi, H. Wang, X. Zhou *et al.*, "Energy and spectrum efficient federated learning via high-precision over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 23, no. 2, pp. 1228–1242, Feb. 2024.

[17] S. Wang, M. Chen, C. Shen, C. Yin, and C. G. Brinton, "Digital over-the-air federated learning in multi-antenna systems," *IEEE Trans. Wireless Commun.*, 2024, early access, doi: 10.1109/TWC.2024.3425732.

[18] J. Yao, W. Xu, Z. Yang, X. You, M. Bennis *et al.*, "Wireless federated learning over resource-constrained networks: Digital versus analog transmissions," *IEEE Trans. Wireless Commun.*, 2024, early access, doi: 10.1109/TWC.2024.3407822.

[19] J. Oh, D. Lee, D. Won, W. Noh, and S. Cho, "Communication-efficient federated learning over-the-air with sparse one-bit quantization," *IEEE Trans. Wireless Commun.*, 2024, early access, doi: 10.1109/TWC.2024.3432758.

[20] Y. Xu, Z. Jiang, H. Xu, Z. Wang, C. Qian *et al.*, "Federated learning with client selection and gradient compression in heterogeneous edge systems," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 5446–5461, May 2024.

[21] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser *et al.*, "Adding gradient noise improves learning for very deep networks," Nov. 2015. [Online]. Available: https://arxiv.org/abs/1511.06807

[22] J. Ren, W. Ni, H. Tian, and G. Nie, "Convergence analysis and latency minimization for semi-federated learning in massive IoT networks," *IEEE Trans. Green Commun. Networking*, vol. 8, no. 1, pp. 413–426, Mar. 2024.

[23] H. H. Yang, Z. Chen, T. Q. S. Quek, and H. V. Poor, "Revisiting analog over-the-air machine learning: The blessing and curse of interference," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 3, pp. 406–419, Apr. 2022.

[24] Z. Chen, H. H. Yang, and T. Q. S. Quek, "Edge intelligence over the air: Two faces of interference in federated learning," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 62–68, Dec. 2023.

[25] Z. Zhang, G. Zhu, R. Wang, V. K. N. Lau, and K. Huang, "Turning channel noise into an accelerator for over-the-air principal component analysis," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7926–7941, Oct. 2022.

[26] N. Huang, M. Dai, Y. Wu, T. Q. S. Quek, and X. Shen, "Wireless federated learning with hybrid local and centralized training: A latency minimization design," *IEEE J. Sel. Top. Signal Process.*, vol. 17, no. 1, pp. 248–263, Jan. 2023.

[27] J. Han, W. Ni, and L. Li, "Semi-federated learning for connected intelligence with computing-heterogeneous devices," *IEEE Internet Things J.*, 2024, early access, doi: 10.1109/JIOT.2024.3355160.

[28] X. Liu, Y. Deng, and T. Mahmoodi, "Wireless distributed learning: A new hybrid split and federated learning approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2650–2665, Apr. 2023.

[29] J. Zheng, H. Tian, W. Ni, G. Nie, W. Jiang *et al.*, "Retransmission-based semi-federated learning," *IEEE Trans. Wireless Commun.*, 2024, early access, doi: 10.1109/TWC.2024.3466177.

[30] 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.

[31] S. S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, "On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2041–2057, Feb. 2022.

[32] 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.

[33] J. Zheng, H. Tian, W. Ni, and P. Zhang, "Supplementary material for the paper: 'Semi-federated learning accelerated by over-the-air distortion'," Oct. 2024. [Online]. Available: https://github.com/zhengjh721/ZJH_over_the_air_distortion_SemiFL

[34] S. Boyd and L. Vandenberghe, *Convex Optimization*. UK: Cambridge University Press, 2004.

[35] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[36] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[37] Z.-q. Luo, W.-k. Ma, A. M.-c. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[38] F. Tian, X. Zhang, X. Wang, and Y.-J. Gong, "Two-layer optimization with utility game and resource control for federated learning in edge networks," *IEEE Trans. Mob. Comput.*, 2024, early access, doi: 10.1109/TMC.2024.3437435.

[39] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document TR 38.901, V16.1.0, 3GPP, Nov. 2020.

[40] W. Ni, Y. Liu, Z. Yang, H. Tian, and X. Shen, "Federated learning in multi-RIS-aided systems," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9608–9624, Jun. 2022.

[41] W. Liu, X. Zang, Y. Li, and B. Vucetic, "Over-the-air computation systems: Optimization, analysis and scaling laws," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5488–5502, Aug. 2020.