Edge Federated Learning Via Unit-Modulus Over-The-Air Computation

Shuai Wang, *Member, IEEE*, Yuncong Hong, *Student Member, IEEE*,
Rui Wang, *Member, IEEE*, Qi Hao, *Member, IEEE*,
Yik-Chung Wu, *Senior Member, IEEE*, and
Derrick Wing Kwan Ng, *Fellow, IEEE*

arXiv:2101.12051v5 [cs.IT] 11 Apr 2022

Abstract

Edge federated learning (FL) is an emerging paradigm that trains a global parametric model from distributed datasets based on wireless communications. This paper proposes a unit-modulus over-the-air computation (UMAirComp) framework to facilitate efficient edge federated learning, which simultaneously uploads local model parameters and updates global model parameters via analog beamforming. The proposed framework avoids sophisticated baseband signal processing, leading to low communication delays and implementation costs. Training loss bounds of UMAirComp FL systems are derived and two low-complexity large-scale optimization algorithms, termed penalty alternating minimization (PAM) and accelerated gradient projection (AGP), are proposed to minimize the nonconvex nonsmooth loss bound. Simulation results show that the proposed UMAirComp framework with PAM algorithm achieves a smaller mean square error of model parameters' estimation, training loss, and test error compared with other benchmark schemes. Moreover, the proposed UMAirComp framework with AGP algorithm achieves satisfactory performance while reduces the computational complexity by orders of magnitude compared with existing optimization algorithms. Finally, we demonstrate the

Part of this paper has been presented at the IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, Dec. 2021 [46].

Shuai Wang is with the Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: s.wang@siat.ac.cn).

Yuncong Hong and Rui Wang are with the Department of Electrical and Electronic Engineering, Southern University of Science and Technology (SUSTech), Shenzhen 518055, China (e-mail: hongyc@mail.sustech.edu.cn; wang.r@sustech.edu.cn).

Qi Hao is with the Department of Computer Science and Engineering and the Sifakis Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology (SUSTech), Shenzhen 518055, China (e-mail: hao.q@sustech.edu.cn).

Yik-Chung Wu is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: ycwu@eee.hku.hk).

Derrick Wing Kwan Ng is with the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia (email: w.k.ng@unsw.edu.au).

implementation of UMAirComp in a vehicle-to-everything autonomous driving simulation platform. It is found that autonomous driving tasks are more sensitive to model parameter errors than other tasks since the neural networks for autonomous driving contain sparser model parameters.

Index Terms

Autonomous driving, federated learning, large-scale optimization, over-the-air computation.

I. Introduction

Deep learning has achieved unprecedented breakthrough in image classification, speech recognition, and object detection due to its ability to efficiently extract intricate nonlinear features from high-dimensional data [1]. Typically, a cloud center collects data from distributed users and trains a centralized model via gradient-based back propagation [2]–[5]. However, since the users need to share their local data to the cloud, this paradigm could lead to some potential privacy issues, hindering the development of deep learning in extensive applications such as smart cities and financial systems.

To address the privacy issue, federated learning (FL), which trains individual deep learning models at users, has been proposed by Google Research [6]. In the framework of FL, the locally generated data is locally adopted and not shared to any third party. To leverage the knowledge from other users, local model parameters are uploaded periodically to a parameter server for model aggregation and the aggregated global parameters are broadcast to the users for further local updates. Therefore, FL achieves distributed training while ensuring data privacy [7].

A. Edge Federated Learning and Related Work

FL was originally developed for wire-line connected systems [8]. To achieve ubiquitous intelligence, a promising solution is edge FL, e.g., [7]–[17], where users are connected to an edge server via wireless links. However, the convergence of edge FL may take a long time due to limited capacity of wireless channels during the uplink model aggregation step. To reduce the transmission delay, various edge FL designs have been proposed (summarized in Table I[1]), which are mainly categorized into digital modulation [8]–[13] and analog modulation [14]–[17] methods.

[1]For more related work on digital and analog federated learning, please refer to [7].

TABLE I: A Comparison of Existing and Proposed Schemes.

Modulation	Work	MIMO	RF Chain	Alg. Complex.	Commun. Delay	Objective Function	AirComp	FL Task
Digital	[8]	✗	+	+	+++	N/A	✗	Classification
	[9], [10]	✗	+	+	++	Loss Bound	✗	Classification
	[12]	✗	+	+	++	Loss Bound	✗	Classification
	[13]	Digital	+++	+++	++	MSE	✗	✗
Analog	[14], [15]	✗	+	+	+	Heuristic	✓	Classification
	[16]	Digital	+++	+++	+	MSE	✓	Classification
	[17]	Digital	+++	++	+	Noise Variance	✓	Classification
	Ours	Analog	+	+	+	Loss Bound	✓	Object Detection

The symbol "+" means low, "++" means moderate, "+++" means high.
The symbol "✓" means functionality supported, "✗" means functionality not supported.

For digital modulation and single-antenna systems, data from different users are multiplexed either in the time or the frequency domain. Current works on delay reduction focus on reducing 1) the number of model aggregation iterations [9], 2) the number of users [10], or 3) the number of bits for representing the gradient of back propagation in each iteration [12]. However, since these strategies involve approximation or simplification of the FL procedure, the performance of learning would be degraded inevitably. Another way to reduce the transmission delay is to adopt multiple-input multiple-output (MIMO) technology for transmission so that data from multiple users are multiplexed concurrently in the spatial domain [13].

On the other hand, the key advantage of analog modulation [14]–[17] over digital modulation arises from the ground-breaking idea of over-the-air computation (AirComp). Specifically, if multiple users upload their local parameters simultaneously, a superimposed signal, which represents a weighted sum of individual model parameters, is observed at the edge server. By performing minimum mean square error (MMSE) detection on the superimposed signal, an estimate of the global parameter vector can be obtained. This significantly saves the transmission time since AirComp in fact exploits inter-user interference in the simultaneous user transmission [18]–[20], in oppose to interference suppression as in digital modulation. Furthermore, the wireless channel can be viewed as a natural privacy-preserving mask for the released models, which injects random channel noises into the parameters [20].

However, due to channel fading and noise in wireless systems, AirComp employed in single-

antenna systems [14], [15] could result in a large error in the estimation of global model parameters at the edge server, leading to slow convergence of FL. As a remedy, adopting MIMO beamforming [16], [17] could reduce the parameter transmission error by aligning the beams carrying the local parameters' information to the same spatial direction. Nonetheless, the current transmit and receive beamforming designs in MIMO AirComp systems involve exceedingly high radio frequency (RF) chain costs and high computational complexities [16], [17], preventing their practical implementation.

In practice, both digital and analog modulation methods share the same goal, i.e., minimizing the training loss function. However, due to the lack of an explicit form of the training loss function with respect to wireless designs, most works focus on other related objective functions such as mean square error (MSE) [16] and noise variance [17]. Recently, the relationship between the training loss function and the wireless designs is derived in [9]–[11], [21], [22]. Nonetheless, the bounds in [9]–[11], [21], [22] are only applicable if the global model parameters are perfectly broadcast to users. For practical cases involving errors in the model broadcast phase, new training loss bounds are required to capture the training performance, which remains an open problem.

B. Summary of Challenges and Contributions

In summary, despite recent exciting development of edge FL techniques, e.g., [8]–[17], a number of technical challenges remain to be overcome, including

1) **Analog beamforming design under federated learning settings**. Analog beamforming with a proper phase shift network design [23], [24] can help reduce implementation costs compared with digital beamfroming in [16], [17], which has not been studied in edge FL systems, yet. Since the analog beamformer is unit-modulus, it introduces a large number of intractable nonconvex and nonsmooth constraints. Furthermore, the analog beamformer in FL systems aims to minimize the FL training loss bound, which is a non-differentiable function of the beamforming design. This is different from the conventional analog beamforming designs in massive MIMO systems that aim to decode individual information from each user [23], [24].
2) **Reduction of beamforming design complexities.** Most beamforming algorithms [13], [16], [17], [24] rely on the execution of the interior point method (IPM). Yet, since IPM involves the inversion of Hessian matrices, these algorithms are with high computational complexities requiring exceedingly long signal processing delay, especially when massive

MIMO technique is applied. On the other hand, first-order methods [38]–[40] (i.e., no inversion of Hessian matrices) could lead to significantly lower computational complexities than that of the IPM. However, they cannot directly handle the analog beamforming design problem due to the non-differentiable training loss bound and unit-modulus constraints.

3) **Verification of robustness in more complex learning tasks**. Existing algorithms in [8]–[17] are mainly tested on simple image classification tasks (e.g., recognition of handwritten digits). Experiments on more complex and closer-to-reality tasks, such as 3D object detection [25], [26] in vehicle-to-everything (V2X) autonomous driving systems, are needed to verify the robustness of edge FL. However, the associated implementation requires a software cluster including scenario generation, multi-modal data generation, multi-sensor calibration, multi-vehicle synchronization and coordinate transformation, label generation, and object detection.

To fill the research gap, this paper proposes the unit-modulus AirComp (UMAirComp) framework for edge FL in MIMO communication systems, as shown in Fig. 1. The UMAirComp framework consists of multiple edge users with local sensing data (e.g., autonomous driving cars generate camera images and light detection and ranging (LiDAR) point clouds of the environment), an edge server for performing model aggregation, and communication interfaces for exchanging model parameters. Specifically, the edge users update their local models using the local training datasets separately. Then, the trained model parameters are transformed into signals via analog modulation and uploaded to the server. To reduce the implementation cost of RF chains, the edge server does not process the received signals at the baseband. Instead, it applies a phase shift network (either a fully-connected structure or a partially-connected structure) in the RF domain for global model aggregation and broadcasting. All users decode model parameters from the received broadcast signals via analog demodulation. The advantages of UMAirComp and contributions of this paper are summarized below:

1) The UMAirComp at the server significantly reduces the required implementation cost of RF chains in MIMO FL systems, thereby reducing the hardware and energy costs. To understand how UMAirComp works, the training loss of UMAirComp framework is proved to be upper bounded by a monotonic increasing function of the maximum MSE of the model parameters' estimation.
2) Despite the UMAirComp problem being highly nonconvex, two large-scale optimization

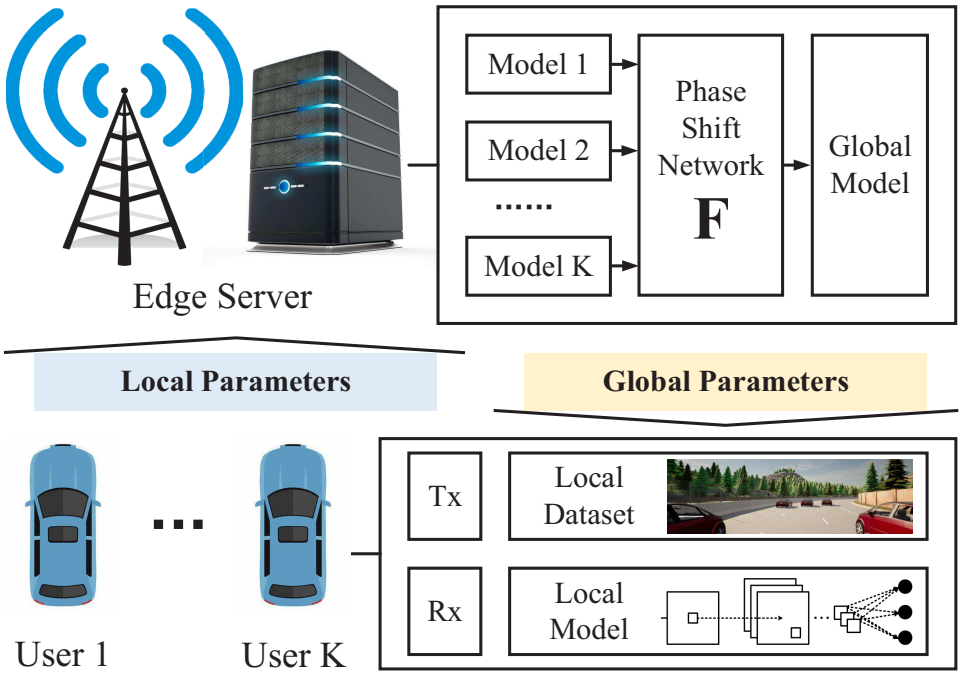


Fig. 1: Illustration of the UMAirComp edge FL system.

algorithms, termed penalty alternating minimization (PAM) and accelerated gradient projection (AGP), are developed for fully-connected UMAirComp and partially-connected UMAirComp, respectively. The learning performance of the proposed PAM and AGP is shown to outperform other benchmark schemes. In particular, the AGP algorithm is 100x faster than that of the state-of-the-art optimization algorithms.

3) We implement the UMAirComp edge FL scheme for 3D object detection with multi-vehicle point-cloud datasets in CARLA simulation platform [27]. To the best of our knowledge, this is the first attempt that edge FL is demonstrated in a V2X auto-driving simulator with a close-to-reality task.

C. Notation

Italic letters, lowercase and uppercase bold letters represent scalars, vectors, and matrices, respectively. Curlicue letters stand for sets and $|\cdot|$ is the cardinality of a set. The operators $\|\cdot\|_2, (\cdot)^T, (\cdot)^H, (\cdot)^{-1}, \lambda_{\max}(\cdot), \lambda_{\min}(\cdot), \mathrm{Null}(\cdot), \mathrm{Rank}(\cdot), \mathrm{vec}(\cdot)$ are the ℓ_2-norm, transpose, Hermitian, inverse, largest eigenvalue, smallest eigenvalue, null space, rank, and vectorization of a matrix, respectively. The operators ∂f and ∇f are the partial derivative and the gradient of

the function f. The function $[x]^+ = \max(x, 0)$, $\mathrm{Re}(x)$ takes the real part of x, $\mathrm{Im}(x)$ takes the imaginary part of x, $\mathrm{conj}(x)$ takes the conjugate of x, and $|x|$ is the modulus of x. $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, $\mathbf{1}_N$ represents the $N \times 1$ all-ones vector, and $\mathbf{A} \succeq \mathbf{B}$ represents $\mathbf{A} - \mathbf{B}$ being positive semidefinite. Finally, $\mathrm{j} = \sqrt{-1}$, $\mathbb{E}(\cdot)$ denotes the expectation of a random variable and $\mathcal{O}(\cdot)$ is the big-O notation standing for the order of arithmetic operations.

II. Edge Federated Learning with AirComp

We consider an edge FL system shown in Fig. 1, which consists of an edge server equipped with N antennas and K single-antenna mobile users. The dataset and model parameter vector at user k are denoted as $\mathcal{D}_k$ and $\mathbf{x}_k \in \mathbb{R}^{M\times 1}$, respectively. Mathematically, the FL procedure aims to solve the following optimization problem:

$$\begin{aligned} \min_{\{\mathbf{x}_k\},\boldsymbol{\theta}} \quad & \underbrace{\frac{1}{\sum_{k=1}^{K}|\mathcal{D}_k|}\sum_{k=1}^{K}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\Theta(\mathbf{d}_{k,l},\boldsymbol{\theta})}_{:=\Lambda(\boldsymbol{\theta})} \\ \text{s.t.} \quad & \mathbf{x}_1 = \cdots = \mathbf{x}_K = \boldsymbol{\theta}, \end{aligned} \tag{1}$$

where $\Theta(\mathbf{d}_{k,l}, \boldsymbol{\theta})$ is the loss function corresponding to a single sample $\mathbf{d}_{k,l}$ $(1 \le l \le |\mathcal{D}_k|)$ in $\mathcal{D}_k$ given parameter vector $\boldsymbol{\theta}$, while $\Lambda(\boldsymbol{\theta})$ denotes the global loss function to be minimized.

The objective function can be rewritten into a separable form $\Lambda(\boldsymbol{\theta}) = \sum_{k=1}^{K}\alpha_k\lambda_k(\mathbf{x}_k)$, where $\lambda_k(\mathbf{x}_k) = \frac{1}{|\mathcal{D}_k|}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\Theta(\mathbf{d}_{k,l}, \mathbf{x}_k)$ is the loss function at the k-th user and $\alpha_k = \frac{|\mathcal{D}_k|}{\sum_{l=1}^{K}|\mathcal{D}_l|}$. Therefore, the training of FL model parameters (i.e., solving (1)) in the considered edge system is naturally a distributed and iterative procedure, where each iteration involves four steps: 1) updating the local parameter vectors $(\mathbf{x}_1, \cdots, \mathbf{x}_K)$ for minimizing $(\lambda_1, \cdots, \lambda_K)$ with respect to $\{\mathcal{D}_1, \cdots, \mathcal{D}_K\}$ at users $(1, \cdots, K)$, respectively; 2) transforming the local parameters $(\mathbf{x}_1, \cdots, \mathbf{x}_K)$ into symbols $(\mathbf{s}_1, \cdots, \mathbf{s}_K)$ for uploading; 3) aggregating $(\mathbf{s}_1, \cdots, \mathbf{s}_K)$ in an analog manner at the edge server and broadcasting the results to the users. 4) receiving the signals $(\mathbf{y}_1, \cdots, \mathbf{y}_K)$ at the users and transforming $(\mathbf{y}_1, \cdots, \mathbf{y}_K)$ into parameters $(\mathbf{x}_1, \cdots, \mathbf{x}_K)$ for next-round updates. The above four steps are elaborated below.

1) Step 1: Let $\mathbf{x}_k^{[i]}(0) \in \mathbb{R}^{M\times 1}$ be the local parameter vector at user k at the beginning of the i-th iteration ($i \ge 0$ and $\mathbf{x}_k^{[0]}(0) = \boldsymbol{\theta}^{[0]}$). To update $\mathbf{x}_k^{[i]}(0)$, user k minimizes the loss

function $\lambda_k(\mathbf{x}_k)$ via gradient descent[2] as

$$\mathbf{x}_k^{[i]}(\tau+1) = \mathbf{x}_k^{[i]}(\tau) - \frac{\varepsilon}{|\mathcal{D}_k|} \sum_{\mathbf{d}_{k,l} \in \mathcal{D}_k} \nabla\Theta\left(\mathbf{d}_{k,l}, \mathbf{x}_k^{[i]}(\tau)\right), \tag{2}$$

where ε is the step-size and τ is from 0 to $E-1$ with E being the number of local updates.

2) Step 2: All users upload $\{\mathbf{x}_k^{[i]}(E)|\forall k\}$ to the edge server. Specifically, user k modulates its local parameter vector $\mathbf{x}_k^{[i]}(E)$ into a complex vector $\mathbf{s}_k^{[i]} \in \mathbb{C}^{S\times 1}$ with $S = M/2$ in an analog manner as in [14], [15], where

$$\mathbf{s}_k^{[i]} = \frac{\sqrt{p_k^{[i]}}\exp(\mathrm{j}\varphi_k^{[i]})}{\sqrt{2\eta^{[i]}}} \left[x_{k,1}^{[i]}(E) + \mathrm{j}\, x_{k,2}^{[i]}(E), \cdots, x_{k,M-1}^{[i]}(E) + \mathrm{j}\, x_{k,M}^{[i]}(E)\right]^T. \tag{3}$$

In (3), $x_{k,m}^{[i]}(E)$ is the m-th element of $\mathbf{x}_k^{[i]}(E)$. The scaling factor $\eta^{[i]} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M}\|\mathbf{x}_k^{[i]}(E)\|_2^2$ such that the average power of $\mathbf{s}_k^{[i]}$ is $\frac{1}{S}\mathbb{E}\left[\|\mathbf{s}_k^{[i]}\|_2^2\right] = p_k^{[i]}$.[3] The transmit power is $p_k^{[i]} \le P_0$ with P_0 being the maximum transmit power at each user and the phase shift is $\varphi_k^{[i]} \in [0, 2\pi]$. To facilitate the subsequent derivations, we define the transmitter design $t_k^{[i]} = \sqrt{p_k^{[i]}}\exp(\mathrm{j}\varphi_k^{[i]})$ for all (i,k) and $\{p_k^{[i]}, \varphi_k^{[i]}\}$ can be recovered from $\{t_k^{[i]}\}$.

3) Step 3: The received signal $\mathbf{R}^{[i]} \in \mathbb{C}^{N\times S}$ at the server is

$$\mathbf{R}^{[i]} = \sum_{k=1}^{K} \mathbf{h}_k^{[i]}(\mathbf{s}_k^{[i]})^T + \mathbf{Z}^{[i]}, \tag{4}$$

where $\mathbf{h}_k^{[i]} \in \mathbb{C}^{N\times 1}$ is the uplink channel vector from user k to the server and $\mathbf{Z}^{[i]} \in \mathbb{C}^{N\times S}$ is the matrix of the additive white Gaussian noise with covariance matrix $\mathbb{E}\left[\mathrm{vec}(\mathbf{Z}^{[i]})\mathrm{vec}(\mathbf{Z}^{[i]})^H\right] = \sigma_b^2\mathbf{I}_{NS}$ (σ_b^2 is the noise power at the server). Upon receiving the superimposed signal, the server processes $\mathbf{R}^{[i]}$ using a function $\Psi^{[i]}(\cdot)$ and broadcasts $\Psi^{[i]}(\mathbf{R}^{[i]}) \in \mathbb{C}^{N\times S}$ to all the users.

4) Step 4: The received signal at user k is

$$\left(\mathbf{y}_k^{[i]}\right)^T = \left(\mathbf{g}_k^{[i]}\right)^H \Psi^{[i]}\left(\mathbf{R}^{[i]}\right) + \left(\mathbf{n}_k^{[i]}\right)^T, \tag{5}$$

where $\mathbf{g}_k^{[i]} \in \mathbb{C}^{N\times 1}$ is the downlink channel vector[4] from the server to user k and $\mathbf{n}_k^{[i]} \in \mathbb{C}^{S\times 1}$ is the vector of the additive white Gaussian noise with covariance matrix $\sigma_k^2\mathbf{I}_S$ (σ_k^2 is the

[2]If $|\mathcal{D}_k|$ is large, stochastic gradient descent can be adopted to accelerate the training speed.

[3]In practice, the users would adopt $\eta^{[i-1]}$ sent from the server in the last iteration, which is a good approximation for the actual $\eta^{[i]}$ in the current iteration.

[4]It is assumed that $\{\mathbf{h}_k^{[i]}, \mathbf{g}_k^{[i]}\}$ are quasi-static flat-fading channels during the model exchange procedure of each FL iteration.

noise power at user k). User k demodulates $\mathbf{y}_k^{[i]}$ and sets the local parameter vector for the $(i+1)$-th iteration as

$$\mathbf{x}_k^{[i+1]}(0) = \sqrt{2\eta^{[i]}}\left[\mathrm{Re}(r_k^{[i]}y_{k,1}^{[i]}), \mathrm{Im}(r_k^{[i]}y_{k,1}^{[i]}), \cdots, \mathrm{Re}(r_k^{[i]}y_{k,S}^{[i]}), \mathrm{Im}(r_k^{[i]}y_{k,S}^{[i]})\right]^T, \tag{6}$$

where $r_k^{[i]} \in \mathbb{C}$ is a receive coefficient applied to $\mathbf{y}_k^{[i]}$ and $y_{k,s}^{[i]}$ is the s-th element of $\mathbf{y}_k^{[i]}$. This completes one federated learning round and we set $i \leftarrow i+1$.

The entire procedure stops when $i = R$ with R being the number of federated learning rounds.

III. Proposed UMAirComp Edge FL Framework

In existing AirComp schemes, e.g., [14]–[17], $\Psi^{[i]}$ is implemented using baseband signal processing techniques. Thus, the superimposed signals of all receive antennas at the edge server are first combined via a vector in the digital baseband and then broadcast to users via another vector in the baseband. Therefore, the associated implementation cost could be high if massive number of antennas are deployed. In the following, the UMAirComp scheme, where the functions $\{\Psi^{[i]}\}$ are implemented in analog domain, is proposed. This helps to reduce both the required implementation costs and power consumptions while achieving excellent system performance.

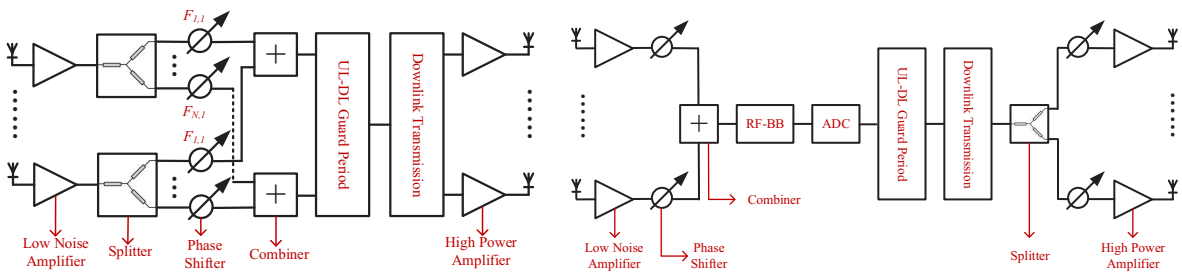
A. UMAirComp Framework

The proposed UMAirComp scheme employs multiple phase shifters to generate the updated global model parameters from the local ones. Specifically, we propose to aggregate and forward the signal via

$$\Psi^{[i]}(\mathbf{R}^{[i]}) = \sqrt{\gamma}\,\mathbf{F}^{[i]}\mathbf{R}^{[i]}, \tag{7}$$

where $\gamma > 0$ is the power scaling factor at the edge server and $\mathbf{F}^{[i]} \in \mathbb{C}^{N\times N}$ is the phase shift matrix. The analog domain implementation of $\mathbf{F}^{[i]}$ is shown in Fig. 2, which inserts an analog beamformer in the RF domain immediately after the low-noise amplifiers and bandpass filter. This is in contrast to the digital domain implementation, which inserts a complete RF receiver chain and a high-rate high-resolution analog-to-digital-converter (ADC) unit for each antenna. The analog beamformer constructs linear combinations of slightly delayed antenna signals and results in N_{ADC} output signals. Depending on the value of N_{ADC}, the implementation can be categorized into fully-connected (Fig. 2a) or partially-connected (Fig. 2b) [23], [24].

For the fully-connected structure shown in Fig. 2a, we have $N_{\mathrm{ADC}} > 1$. Each entry $F_{l,l'}^{[i]}$ of $\mathbf{F}^{[i]}$ corresponds to the phase delay introduced by the phase shifter for the l-th receive antenna and



(a) (b)

Fig. 2: Illustration of the phase shift network at the server: a) fully-connected structure; b) partially-connected structure.

the l'-th output signal. Note that amplitude changes are not possible. This results in unit-modulus constraints on all elements of the analog beamforming matrices $\{\mathbf{F}^{[i]}\}$, i.e.,

$$|F_{l,l'}^{[i]}| = 1, \quad \forall l, l'. \tag{8}$$

The output signals are stacked into vectors, delayed for a guard period (e.g., using RF sampling or RF delay line systems), converted to downlink signals, amplified by γ, and transmitted to users.

For the partially-connected structure shown in Fig. 2b, we have $N_{\text{ADC}} = 1$. Each receive antenna element is processed by a phase shifter and combined into a single output signal. The output signal is downconverted to a baseband signal, which is sampled and quantized into bits via an ADC unit. After delaying for a guard period, the signal is upconverted to a passband signal and filtered via another phase shift vector before transmission. Hence, apart from the unit-modulus constraint (8), $\mathbf{F}^{[i]}$ should also satisfy $\text{Rank}\left(\mathbf{F}^{[i]}\right) = 1$. Note that the required number of total phase shifters is only $2N$ for the partially-connected structure.

Based on equations (2)–(7), the MSE between the received local parameter $\mathbf{x}_k^{[i+1]}$ and the target local parameter $\boldsymbol{\theta}^{[i]}$ at the $(i+1)$-th FL iteration for the proposed UMAirComp scheme is

$$\begin{aligned}\mathbb{MSE}_k^{[i]}\left(\mathbf{F}^{[i]}, \{t_k^{[i]}, r_k^{[i]}\}\right) &= \mathbb{E}\left[\left\|\mathbf{x}_k^{[i+1]}(0) - \boldsymbol{\theta}^{[i]}\right\|_2^2\right] \\ &= 2S\eta^{[i]}\left[\gamma\sum_{j=1}^{K}\left|r_k^{[i]}(\mathbf{g}_k^{[i]})^H\mathbf{F}^{[i]}\mathbf{h}_j^{[i]}t_j^{[i]} - \alpha_j\right|^2 + \gamma\sigma_b^2\|r_k^{[i]}(\mathbf{g}_k^{[i]})^H\mathbf{F}\|_2^2 + \sigma_k^2|r_k^{[i]}|^2\right],\end{aligned} \tag{9}$$

where the target model parameter

$$\boldsymbol{\theta}^{[i]} = \sum_{k=1}^{K}\alpha_k\mathbf{x}_k^{[i]}(E) \tag{10}$$

is equivalent to the gradient descent of the objective function of (1). Note that the $\mathbb{MSE}_k^{[i]}$ is the total MSE between the obtained and target models, which consists of $M = 2S$ parameters. In the following, we will also use $\mathbb{MSE}_k^{[i]}/(2S\eta^{[i]})$ as a performance metric, which is the normalized MSE representing the relative distortion with respect to each parameter.

B. Training Loss Analysis

Due to the model distortion $\mathbb{MSE}_k^{[i]}$ in (9), the training loss $\Lambda(\mathbf{x}_k^{[R]}(0))$ in (1) after the UMAirComp edge FL terminates is random. Its expectation $\mathbb{E}[\Lambda(\mathbf{x}_k^{[R]}(0))]$, where the expectation is taken over channel noises and model parameters, would be greater than $\Lambda(\boldsymbol{\theta}^*)$ in wired FL systems, where $\boldsymbol{\theta}^*$ denotes the optimal solution of $\boldsymbol{\theta}$ to (1). Hence, we can use their difference, namely $\mathbb{E}[\Lambda(\mathbf{x}_k^{[R]}(0))] - \Lambda(\boldsymbol{\theta}^*)$, as a metric to capture the degradation on training loss. Before establishing the main results, we first introduce the assumption imposed on the loss function.

Assumption 1. *(i) The function $\Lambda(\mathbf{x})$ is μ-strongly convex. (ii) The function $\frac{1}{|\mathcal{D}_k|}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\Theta(\mathbf{d}_{k,l},\mathbf{x})$ is twice differentiable and satisfies $\frac{1}{|\mathcal{D}_k|}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\nabla_{\mathbf{x}}^2\Theta(\mathbf{d}_{k,l},\mathbf{x}) \preceq L\mathbf{I}$.*

Under Assumption 1, the relationship between $\Lambda(\mathbf{x}_k^{[R]}(0))$ and $\Lambda(\boldsymbol{\theta}^*)$ is summarized in the following theorem.

Theorem 1. *With $E = 1$ and $\varepsilon = \frac{1}{L}$, the UMAirComp scheme satisfies*

$$\mathbb{E}\left[\Lambda(\mathbf{x}_k^{[R]}(0))\right] - \Lambda(\boldsymbol{\theta}^*) \leq \sum_{i=0}^{R-1} A^{[i]} \max_{k=1,\cdots,K} \mathbb{MSE}_k^{[i]}, \tag{11}$$

as $R \to +\infty$, where

$$A^{[i]} = \frac{L}{2}\left(3 + 2\sum_{j=1}^{K}\alpha_j^2\right)\left(1 - \frac{\mu}{L}\right)^{R-1-i}. \tag{12}$$

Proof. First, it can be proved that $\mathbb{E}\left[\|\Delta\mathbf{x}^{[i]}\|_2^2\right] \leq \left(3 + 2\sum_{j=1}^{K}\alpha_j^2\right)\max_k \mathbb{MSE}_k^{[i]}$, where $\Delta\mathbf{x}^{[i]} = \mathbf{x}_k^{[i+1]}(0) - \left[\mathbf{x}_k^{[i]}(0) - \varepsilon\nabla_{\mathbf{x}}\Lambda(\mathbf{x}_k^{[i]}(0))\right]$. Second, using Assumption 1 and Lipschitz conditions, the relationship between $\Lambda(\mathbf{x}_k^{[i+1]}(0)) - \Lambda(\mathbf{x}_k^{[i]}(0))$ and $\mathbb{E}\left[\|\Delta\mathbf{x}^{[i]}\|_2^2\right]$ is obtained. Lastly, applying the former relationship recursively, the relationship between $\mathbb{E}[\Lambda(\mathbf{x}_k^{[i+1]}(0))] - \Lambda(\boldsymbol{\theta}^*)$ and $\mathbb{E}\left[\|\Delta\mathbf{x}^{[i]}\|_2^2\right]$ is obtained. More details can be found in Appendix A. ■

Theorem 1 shows a diminishing $A^{[i]} \to 0$ for a large $R - 1 - i$, meaning that the impact from earlier FL iterations vanishes as the edge FL continues. On the other hand, if $\mathbb{MSE}_k^{[i]} \to 0$

for all k, then $\Lambda(\mathbf{x}_k^{[R]})$ is an unbiased estimate of $\Lambda(\boldsymbol{\theta}^*)$. This demonstrates the effectiveness of UMAirComp in the asymptotic region. The convexity and smoothness in Assumption 1 have been adopted in most loss bound analysis of FL (e.g., [21], [22], [28], [40]). Although it seems to be restrictive for realistic applications, analysis under Assumption 1 could provide important insights of the behavior of UMAirComp in nonconvex cases.

The analysis can be generalized to the case with multiple local epochs (i.e., $E > 1$). Specifically, in addition to Assumption 1, we further impose the following assumption.

Assumption 2. $\|\frac{1}{|\mathcal{D}_k|}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\nabla_{\mathbf{x}}\Theta(\mathbf{d}_{k,l},\mathbf{x})\|_2^2 \leq G^2$ *for some* $G > 0$ *and all* k.

In practice, the above assumption can always be satisfied with a sufficiently large G, which represents the maximum gradient norm in the back propagation procedure. Otherwise, the exploding gradients will lead to the unstable training of deep networks. Let Λ^* denotes the minimum training loss in (1) over all users' datasets and λ_k^* denotes the minimum training loss over the k-th user's dataset. Then, the degree of heterogeneity of the edge FL system is defined as $\Gamma = \Lambda^* - \sum_{k=1}^K \alpha_k \lambda_k^*$. It can be seen that $\Gamma = 0$ if all users have the same dataset. Then the following convergence result can be established.

Theorem 2. *With* $\varepsilon = \frac{2}{\mu(\nu+iE+\tau)}$ *and* $\nu = \max(\frac{8L}{\mu}, E)$*, the UMAirComp scheme satisfies*

$$\mathbb{E}\left[\Lambda(\mathbf{x}_k^{[R]}(0))\right] - \Lambda(\boldsymbol{\theta}^*) \leq \frac{2L\max(4C, \mu^2\nu\|\boldsymbol{\theta}^{[0]} - \boldsymbol{\theta}^*\|_2^2)}{\mu^2(RE+\nu)}, \tag{13}$$

where

$$C = 8E^2G^2 + 6L\Gamma + \frac{\mu^2(\nu+RE)^2}{4}\max_{\forall i,k}\mathbb{MSE}_k^{[i]}. \tag{14}$$

Proof. First, we define a virtual sequence and a recursive bound can be obtained for this sequence based on [21, Lemma 1]. Second, the recursive bound is further simplified by bounding the model distortion and gradient deviation. Lastly, a non-recursive bound is obtained using the induction method in [22, Appendix B] and the results follow immediately. More details can be found in Appendix B. ■

It can be seen from Theorem 2 that if we increase the number of local epochs E or the degree of heterogeneity Γ, the term C would increase and the gap between $\mathbb{E}\left[\Lambda(\mathbf{x}_k^{[R]}(0))\right]$ and $\Lambda(\boldsymbol{\theta}^*)$ becomes larger. This implies that more local training epochs or more diverse datasets would make it more difficult for the proposed UMAirComp federated learning to converge.

C. Problem Formulation

Ideally, the optimization of $\{\mathbf{F}^{[i]}, r_k^{[i]}, t_k^{[i]}\}$ should be performed to minimize the expectation of training error, i.e., $\mathbb{E}[\Lambda(\mathbf{x}_k^{[R]}(0))]$. However, its analytical expression is usually challenging to derive. As a compromise, we resort to the minimization of the upper bound obtained from Theorem 1 or Theorem 2, which at least guarantees the worst-case training loss performance. Minimizing the training loss bound is equivalent to minimizing $\sum_{i=0}^{R} A^{[i]} \max_k \mathbb{MSE}_k^{[i]}$ for the case of $E = 1$ and $\max_{\forall i,k} \mathbb{MSE}_k^{[i]}$ for the case of $E > 1$. But no matter $E = 1$ or $E > 1$, the objective function can be decoupled for each iteration and the minimization at the i-th FL iteration, $\forall i$, is given by

$$\mathcal{P}: \min_{\mathbf{F}, \{r_k, t_k\}} \quad \max_{k=1,\cdots,K} \left[\gamma \sum_{j=1}^{K} \left| r_k \mathbf{g}_k^H \mathbf{F} \mathbf{h}_j t_j - \alpha_j \right|^2 + \gamma \sigma_b^2 \| r_k (\mathbf{g}_k)^H \mathbf{F} \|_2^2 + \sigma_k^2 |r_k|^2 \right] \tag{15a}$$

$$\text{s.t.} \quad \mathbf{F} \in \mathcal{F}, \tag{15b}$$

$$|t_k|^2 \leq P_0, \quad k = 1, \cdots, K, \tag{15c}$$

where the FL iteration index i is omitted. The constraint (15b) is the beamforming constraints at the server, with the feasible set

$$\mathcal{F} = \begin{cases} \{\mathbf{F} : |F_{l,l'}| = 1, \forall l, l'\}, & \text{FC} \\ \{\mathbf{F} : \text{Rank}\,(\mathbf{F}) = 1, \quad |F_{l,l'}| = 1, \forall l, l'\}, & \text{PC} \end{cases}, \tag{16}$$

where FC represents the fully connected case and PC represents the partially connected case.

The constraint (15c) is the power constraints at users obtained from Section II. It can be seen that the key to minimizing the training loss upper bound of UMAirComp is to minimize the *maximum MSE* instead of the average MSE. Problem $\mathcal{P}$ is NP-hard due to the unit-modulus constraints [23], [24]. In addition, the coupling between variables $\{r_k\}$, $\{t_k\}$, and $\mathbf{F}$ makes the problem nonlinear and nonconvex. Furthermore, the large dimensions of $\mathbf{F}$ and $\{r_k, t_k\}$ call for low-complexity algorithms in the scenario with massive antennas and users.

D. Practical Implementation

In practice, parameters $\gamma, \{\alpha_k\}, \sigma_b^2, \{\sigma_k^2\}, P_0, \{\mathbf{h}_k\}, \{\mathbf{g}_k\}$ should be collected at the edge server before solving $\mathcal{P}$. These information can be categorized into two types: system state information and channel state information. The system state information includes the power scaling factor γ at the server, parameter aggregation weights $\{\alpha_k\}$ at the server, noise powers $\{\sigma_b^2, \sigma_k^2\}$ at the

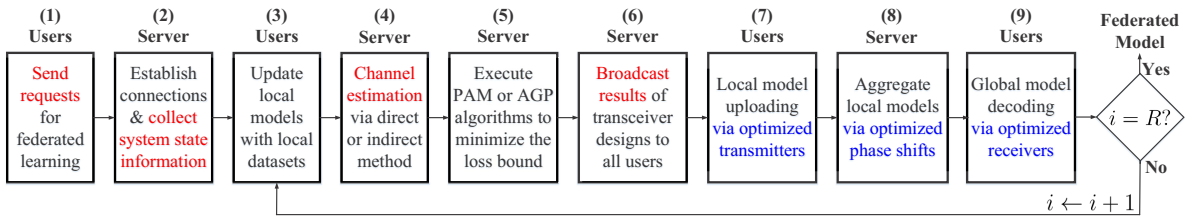


Fig. 3: Flowchart of the proposed UMAirComp federated learning including system state information estimation, local training, channel state information estimation, results broadcast, model uploading and aggregation. The text in red represents the information exchange procedure and the text in blue represents the signal processing procedure.

server and users, and maximum transmit power P_0 at users. This information can be viewed as constants during the entire federated learning procedure. They are collected only once before the first federated learning iteration and pre-stored at the edge server for subsequent use. On the other hand, the channel state information, including uplink channels $\{\mathbf{h}_k\}$ and downlink channels $\{\mathbf{g}_k\}$, needs to be estimated using pilot signals before each federated learning round. Since the digital baseband processor does not have access to the individual antenna signals, the channels in UMAirComp systems should be estimated using low-rate low-resolution ADCs or the least squares fitting [24, Section VI].

Based on the above discussions, the entire signalling procedure for obtaining the necessary information to solve the UMAirComp FL problem is shown in Fig. 3. It can be seen that there are four messages (highlighted in red) that have to be exchanged between the server and users:

1) Users to server: requests for the federated learning;
2) Users to server: system state information including the number of antennas and users, transmit power budgets, sizes of datasets, noise powers;
3) Users to server: pilot signals for channel estimation;
4) Server to users: solutions of transceivers generated by the optimization algorithms.

IV. Penalty Alternating Minimization for Fully-Connected UMAirComp

In this section, the UMAirComp with fully-connected structure is studied, where the feasible set $\mathcal{F}$ is given in the first line of (16). A PAM algorithm with two layers of iterations, i.e., an outer-layer iteration and an inner-layer iteration, will be proposed to optimize the system performance. Below we first introduce the outer-layer iteration.

A. Outer-Layer Iteration

To resolve the coupling between variables $\{r_k\}$, $\{t_k\}$, and $\mathbf{F}$, this paper adopts an alternating optimization framework [29], which optimizes one design variable at a time with others being fixed. Starting with an initial solution $\{\mathbf{F}^{(0)}, r_k^{(0)}, t_k^{(0)}\}$, the entire procedure for solving the problem $\mathcal{P}$ for the $(n+1)$-th outer iteration, $\forall n$, can be elaborated below:

$$\mathbf{F}^{(n+1)} = \arg\min_{\mathbf{F}} \max_{\forall k} \left(\sum_{j=1}^{K} \left| r_k^{(n)} \mathbf{g}_k^H \mathbf{F} \mathbf{h}_j t_j^{(n)} - \alpha_j \right|^2 + \sigma_b^2 \| r_k^{(n)} \mathbf{g}_k^H \mathbf{F} \|_2^2 \right)$$

$$\text{s.t.} \quad |F_{l,l'}| = 1, \quad \forall, l, l', \tag{17a}$$

$$\{r_k^{(n+1)}\} = \arg\min_{\{r_k\}} \max_{\forall k} \left(\gamma \sum_{j=1}^{K} \left| r_k \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j^{(n)} - \alpha_j \right|^2 + \gamma \sigma_b^2 \| r_k \mathbf{g}_k^H \mathbf{F}^{(n+1)} \|_2^2 + \sigma_k^2 |r_k|^2 \right), \tag{17b}$$

$$\{t_k^{(n+1)}\} = \arg\min_{\{t_k\}} \max_{\forall k} \sum_{j=1}^{K} \left| r_k^{(n+1)} \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j - \alpha_j \right|^2$$

$$\text{s.t.} \quad |t_k|^2 \leq P_0, \quad k = 1, \cdots, K, \tag{17c}$$

where $\{\mathbf{F}^{(n)}, t_k^{(n)}, r_k^{(n)}\}$ is the solution at the n-th outer iteration. The iterative procedure stops until n reaches the maximum iteration number $n = N_{\max}$.

Problem (17a) can be transferred to a convex problem via semidefinite relaxation (SDR) while problems (17b) and (17c) are convex. Hence, problems (17a)–(17c) can all be solved via CVX [32], a Matlab software package for solving convex problems based on IPM. According to [32], the computational complexity is at least $\mathcal{O}(N^7)$ for solving (17a) (the vectorization of $\mathbf{F}$ involves N^2 variables) and $\mathcal{O}(K^{3.5})$ for solving (17b)–(17c). For large N and K, this method is not desirable. In the following, a new algorithm termed PAM, which decomposes (17a)–(17c) into smaller subproblems that are either solved by gradient updates or closed-form updates, is proposed for achieving both excellent performance and significantly lower computational complexities.

B. Inner-Layer Iteration

1) Optimization of $\mathbf{F}$*:* First, vectorization of $\mathbf{F}$ is given as $\mathbf{f} = \mathrm{vec}(\mathbf{F}) \in \mathbb{C}^{N^2 \times 1}$. Applying $\mathrm{Tr}\left(\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{X}^T\right) = \mathrm{vec}(\mathbf{X})^T \left(\mathbf{B}^T \otimes \mathbf{A}\right) \mathrm{vec}(\mathbf{X})$ [30], we have

$$r_k^{(n)} \mathbf{g}_k^H \mathbf{F} \mathbf{h}_j t_j^{(n)} = r_k^{(n)} t_j^{(n)} \left(\mathbf{h}_j^T \otimes \mathbf{g}_k^H\right) \mathrm{vec}(\mathbf{F})$$

$$= (\mathbf{a}_{k,j}^{(n)})^H \mathbf{f}, \tag{18}$$

$$\sigma_b^2 \|r_k^{(n)} \mathbf{g}_k^H \mathbf{F}\|_2^2 = |r_k^{(n)}|^2 \mathrm{Tr}\left(\mathbf{g}_k \mathbf{g}_k^H \mathbf{F} \mathbf{I}_N \mathbf{F}^H\right)$$

$$= \mathbf{f}^H \mathbf{G}_k^{(n)} \mathbf{f}, \tag{19}$$

where $\mathbf{a}_{k,j}^{(n)} = \left[r_k^{(n)} t_j^{(n)} \left(\mathbf{h}_j^T \otimes \mathbf{g}_k^H\right)\right]^H$ and $\mathbf{G}_k^{(n)} = \sigma_b^2 |r_k^{(n)}|^2 \mathbf{I}_N \otimes \left(\mathbf{g}_k \mathbf{g}_k^H\right)$. Problem (17a) is thus re-formulated as

$$\mathcal{P}_F : \min_{\mathbf{f}} \quad \max_{k=1,\cdots,K} \quad \left(\sum_{j=1}^{K} \left|(\mathbf{a}_{k,j}^{(n)})^H \mathbf{f} - \alpha_j\right|^2 + \mathbf{f}^H \mathbf{G}_k^{(n)} \mathbf{f}\right)$$

$$\text{s.t.} \quad |f_l| = 1, \quad l = 1, \cdots, N^2. \tag{20}$$

Next, to handle the nonseparable objective function, variable splitting of $\mathbf{f}$ is proposed such that $\mathbf{f} = \mathbf{u}_1 = \cdots = \mathbf{u}_K$, where $\{\mathbf{u}_k\}$ are auxiliary variables. Moreover, to handle the unit-modulus constraints, another auxilliary variable $\mathbf{z} = \mathbf{f}$ is introduced. For all the newly introduced equality constraints, they can be transformed into quadratic penalties in the objective function [31]. As a result, $\mathcal{P}_F$ is transformed into

$$\min_{\mathbf{f},\mathbf{z},\{\mathbf{u}_k\}} \quad \rho\left(\frac{1}{K}\sum_{j=1}^{K} \|\mathbf{u}_j - \mathbf{f}\|_2^2 + \|\mathbf{z} - \mathbf{f}\|_2^2\right) + \max_{\forall k} \quad \left(\sum_{j=1}^{K} \left|(\mathbf{a}_{k,j}^{(n)})^H \mathbf{u}_k - \alpha_j\right|^2 + \mathbf{u}_k^H \mathbf{G}_k^{(n)} \mathbf{u}_k\right)$$

$$\text{s.t.} \quad |z_l| = 1, \quad l = 1, \cdots, N^2, \tag{21}$$

where ρ is a tuning parameter. It can be proved that $\mathcal{P}_F$ and (21) are equivalent problems as $\rho \to +\infty$ [31]. However, this case also leads to the gradient norm of the objective function of (21) being infinite, making (21) difficult to solve. Therefore, ρ controls the tradeoff between approximation error and difficulty in solving (21).

We address (21) using alternating minimization, in which the cost function is iteratively minimized with respect to one variable whereas the others are fixed. Starting with an initial $\mathbf{f}^{(0)} = \mathbf{z}^{(0)} = \mathbf{u}_k^{(0)} = \mathrm{vec}\left(\mathbf{F}^{(n)}\right)$, the whole process consists of iteratively solving

$$\mathbf{u}_k^{(m+1)} = \arg\min_{\mathbf{u}_k} \quad \sum_{j=1}^{K} \left|(\mathbf{a}_{k,j}^{(n)})^H \mathbf{u}_k - \alpha_j\right|^2 + \mathbf{u}_k^H \mathbf{G}_k^{(n)} \mathbf{u}_k + \frac{\rho}{K}\|\mathbf{u}_k - \mathbf{f}^{(m)}\|_2^2, \quad \forall k, \tag{22a}$$

$$\mathbf{f}^{(m+1)} = \arg\min_{\mathbf{f}} \quad \frac{1}{K}\sum_{j=1}^{K} \|\mathbf{u}_j^{(m+1)} - \mathbf{f}\|_2^2 + \|\mathbf{z}^{(m)} - \mathbf{f}\|_2^2, \tag{22b}$$

$$\mathbf{z}^{(m+1)} = \arg\min_{|z_l|=1,\forall l} \quad \rho\|\mathbf{z} - \mathbf{f}^{(m+1)}\|_2^2, \tag{22c}$$

where m is the inner iteration index. It can be verified that the objective function of (21) is strongly convex. Therefore, despite the non-differentiability of the objective, the alternating minimization (22a)–(22c) is guaranteed to converge to a stationary point of (21) [29]. The iterative procedure stops until m reaches the maximum iteration number $m = M_{\max}$.

The remaining question is how to solve (22a)–(22c) optimally. We notice that problems (22a) and (22b) are standard least squares problems, thus their solutions are given by the following closed-form expressions

$$\mathbf{u}_k^{(m+1)} = \left(\sum_{j=1}^{K} \mathbf{a}_{k,j}^{(n)} (\mathbf{a}_{k,j}^{(n)})^H + \mathbf{G}_k^{(n)} + \rho \mathbf{I} \right)^{-1} \times \left(\sum_{j=1}^{K} \alpha_j \mathbf{a}_{k,j}^{(n)} + \frac{\rho}{K} \mathbf{f}^{(m)} \right), \tag{23}$$

$$\mathbf{f}^{(m+1)} = \frac{1}{2} \left(\frac{1}{K} \sum_{j=1}^{K} \mathbf{u}_j^{(m+1)} + \mathbf{z}^{(m)} \right), \tag{24}$$

respectively. On the other hand, problem (22c) is the projection of $\mathbf{f}^{(m+1)}$ onto unit-modulus constraint and the optimal solution is simply

$$\mathbf{z}^{(m+1)} = \exp\left(\mathrm{j} \angle \mathbf{f}^{(m+1)} \right). \tag{25}$$

2) Optimization of $\{r_k\}$*:* The problem of optimizing $\{r_k\}$ in (17b) is also a least squares problem. The optimal solution is found by setting the derivative $\partial \mathbb{MSE}_k / \partial \mathrm{conj}(r_k)$ to zero:

$$\begin{aligned} \frac{\partial \mathbb{MSE}_k}{\partial \mathrm{conj}(r_k)} = \gamma \sum_{j=1}^{K} \left(r_k \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j^{(n)} - \alpha_j \right) \mathrm{conj} \left(\mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j^{(n)} \right) \\ + \gamma \sigma_b^2 \| \mathbf{g}_k^H \mathbf{F}^{(n+1)} \|_2^2 r_k + \sigma_k^2 r_k = 0, \end{aligned} \tag{26}$$

which yields

$$r_k^{(n+1)} = \frac{\sum_j \alpha_j \mathrm{conj} \left(\mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j^{(n)} \right)}{\sum_j |\mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j t_j^{(n)}|^2 + \sigma_b^2 \| \mathbf{g}_k^H \mathbf{F}^{(n+1)} \|_2^2 + \sigma_k^2 / \gamma}. \tag{27}$$

3) Optimization of $\{t_k\}$*:* The objective function of problem (17c) is not separable. Following similar variable splitting procedure as in (21), we introduce auxiliary variables $\{\xi_{k,j} = t_k, \forall k, j\}$ and add quadratic penalty $\rho \sum_{k=1}^{K} \sum_{j=1}^{K} |\xi_{k,j} - t_k|^2$ to the objective function. The problem (17c) is transformed into

$$\begin{aligned} \mathcal{P}_t : \min_{\{t_k, \xi_{k,j}\}} \quad & \rho \sum_{k=1}^{K} \sum_{j=1}^{K} |\xi_{k,j} - t_j|^2 + \max_{\forall k} \ \sum_{j=1}^{K} \left| r_k^{(n+1)} \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j \xi_{k,j} - \alpha_j \right|^2 \\ \text{s.t.} \quad & |t_k|^2 \le P_0, \quad k = 1, \cdots, K, \end{aligned} \tag{28}$$

and variables t_k and $\xi_{k,j}$ can be optimized iteratively. In particular, starting from $t_k^{(0)} = \xi_{k,j}^{(0)} = t_k^{(n)}$, the solutions of $\xi_{k,j}$ and t_k at the q-th iteration are given by

$$\xi_{k,j}^{(q+1)} = \arg\min_{\xi_{k,j}} \left| r_k^{(n+1)} \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j \xi_{k,j} - \alpha_j \right|^2 + \rho |\xi_{k,j} - t_j^{(q)}|^2, \quad \forall k, j, \tag{29}$$

$$t_j^{(q+1)} = \arg\min_{|t_j|^2 \leq P_0} \ \rho \sum_{k=1}^{K} |\xi_{k,j}^{(q+1)} - t_j|^2, \quad \forall j. \tag{30}$$

Problem (29) is a least squares problem and (30) is a quadratic problem with only one constraint. They can be solved optimally based on Karush-Kuhn-Tucker (KKT) conditions and the solutions are given by

$$\xi_{k,j}^{(q+1)} = \frac{\mathrm{conj}(r_k^{(n+1)} \mathbf{g}_k^H \mathbf{F} \mathbf{h}_j) \alpha_j + \rho t_j^{(q)}}{|r_k^{(n+1)} \mathbf{g}_k^H \mathbf{F}^{(n+1)} \mathbf{h}_j|^2 + \rho}, \quad \forall k, j, \tag{31}$$

$$t_j^{(q+1)} = \sqrt{\min(P_0, 1)} \ \frac{\frac{1}{K} \sum_{k=1}^{K} \xi_{k,j}^{(q+1)}}{\left| \frac{1}{K} \sum_{j=1}^{K} \xi_{k,j}^{(q+1)} \right|}, \quad \forall j. \tag{32}$$

The iterative procedure stops until q reaches the maximum iteration number $q = Q_{\max}$.

C. Summary and Complexity Analysis of PAM

In summary, the complete PAM algorithm for solving problem $\mathcal{P}$ with a fully-connected structure consists of two layers of iterations. Let $N_{\max}$ and $M_{\max}$ denote the maximum number of iterations for outer and inner layers, respectively. In the outer layer, the PAM optimizes $\mathbf{F}$, $\{r_k\}$ and $\{t_k\}$ alternatively in each of the $N_{\max}$ iterations. In the inner layer, $\mathbf{F}$ is obtained via computing (23)–(25) for $M_{\max}$ iterations, $\{r_k\}$ is obtained via computing (27), and $\{t_k\}$ is obtained via computing (31)–(32) for $Q_{\max}$ iterations. The computational complexities for these equations are $\mathcal{O}(KN^2)$, $\mathcal{O}(KN^2)$, $\mathcal{O}(N^2)$, $\mathcal{O}(KN^2)$, $\mathcal{O}(K^2)$, $\mathcal{O}(K^2)$ for (23), (24), (25), (27), (31), (32), respectively. Since the computation is dominant by $\mathcal{O}(KN^2)$, the total computational complexity of PAM is $\mathcal{O}(N_{\max} Q_{\max} K N^2)$.

V. Accelerated Gradient Projection for Partially-Connected UMAirComp

In practice, it is possible that there are a large number of antennas at the edge server. In such a case, a smaller number of phase shifters than N^2 at the edge server is desirable. To this end, this section proposes an accelerated gradient projection method for partially-connected UMAirComp, which only needs $2N$ phase shifters.

With a partially-connected structure as illustrated in Fig. 2b, the feasible set $\mathcal{F}$ equals the second line of (16). Since the rank of $\mathbf{F}$ is 1, we can apply rank-one decomposition on $\mathbf{F}$ which yields $\mathbf{F} = \mathbf{v}\mathbf{w}^H$. Then, we adopt the following approximations to $\mathcal{P}$: 1) Set $r_k = 1/(\mathbf{g}_k^H\mathbf{v})$ and $t_k = 1/(\mathbf{h}_k^H\mathbf{w})$; 2) Relax $\{|F_{l,l'}| = 1 | \forall l, l'\}$ into $\|\mathbf{F}\|_2^2 \leq N^2$. After the above steps, problem $\mathcal{P}$ is simplified into a bilevel form:

$$\mathcal{Q}: \min_{\mathbf{v}} \quad \gamma\sigma_b^2\|\mathbf{w}\|_2^2 + \max_{k=1,\cdots,K} \frac{\sigma_k^2}{|\mathbf{g}_k^H\mathbf{v}|^2} \tag{33a}$$

$$\text{s.t.} \quad \frac{N^2}{\|\mathbf{v}\|_2^2} \geq \min_{\mathbf{w}} \left\{ \|\mathbf{w}\|_2^2 : |\mathbf{w}^H\mathbf{h}_k|^2 \geq \frac{\alpha_k^2}{P_0}, \quad \forall k \right\}. \tag{33b}$$

Note that the adopted approximations above $\mathcal{Q}$ would lead to performance loss. However, the solution obtained by solving $\mathcal{Q}$ is asymptotically optimal to $\mathcal{P}$ as $\sigma_k^2, \sigma_b^2 \to 0$. This is because $\gamma\sum_{j=1}^K |r_k\mathbf{g}_k^H\mathbf{F}\mathbf{h}_j t_j - \alpha_j|^2 = 0$ with $\mathbf{F} = \mathbf{v}\mathbf{w}^H$, $r_k = 1/(\mathbf{g}_k^H\mathbf{v})$, and $t_k = \alpha_k/(\mathbf{h}_k^H\mathbf{w})$. Consequently, as noise powers $\sigma_b^2, \sigma_k^2 \to 0$, the objective function of $\mathcal{P}$ with $\{\mathbf{F} = \mathbf{v}\mathbf{w}^H, r_k = 1/(\mathbf{g}_k^H\mathbf{v}), t_k = \alpha_k/(\mathbf{h}_k^H\mathbf{w})\}$ becomes zero, meaning that this solution is optimal to $\mathcal{P}$ in the asymptotic case.

Since the right hand side of the constraint in (33b) is a quadratic optimization problem, it can be solved by the accelerated random coordinate descent method with a complexity of $\mathcal{O}(KN)$ [33]. Denoting the solution of $\mathbf{w}$ as $\mathbf{w} = \mathbf{w}^\diamond$, problem $\mathcal{Q}$ is reduced to

$$\mathcal{Q}_1: \min_{\mathbf{v}} \quad \max_{k=1,\cdots,K} \frac{\sigma_k^2}{|\mathbf{g}_k^H\mathbf{v}|^2} \quad \text{s.t.} \quad \|\mathbf{v}\|_2^2 \leq \beta, \tag{34}$$

where $\beta = \frac{N^2}{\|\mathbf{w}^\diamond\|_2^2}$ and we have removed the term $\gamma\sigma_b^2\|\mathbf{w}^\diamond\|_2^2$ in the objective function (as it is a constant now). In the following, we propose an efficient fixed point method for solving problem $\mathcal{Q}_1$.

A. Fixed-Point Iteration

Solving problem $\mathcal{Q}_1$ is challenging due to 1) the large dimension of variables, and 2) the large number of elements inside the maximum operator. To this end, we first re-write $\mathcal{Q}_1$ as a bilevel problem

$$\begin{aligned} \min_{\|\mathbf{v}\|_2^2 \leq \beta} \max_{k=1,\cdots,K} \frac{\sigma_k^2}{|\mathbf{g}_k^H\mathbf{v}|^2} &\Longleftrightarrow \min_{\|\mathbf{v}\|_2^2 \leq \beta} \max_{k=1,\cdots,K} -\frac{|\mathbf{g}_k^H\mathbf{v}|^2}{\sigma_k^2} \\ &\Longleftrightarrow \min_{\|\mathbf{v}\|_2^2 \leq \beta} \max_{\mathbf{b}\in\Delta} \underbrace{-\sum_{k=1}^K \frac{b_k|\mathbf{g}_k^H\mathbf{v}|^2}{\sigma_k^2}}_{:=h(\mathbf{v},\mathbf{b})}, \end{aligned} \tag{35}$$

where $\Delta = \{\mathbf{b}|\mathbf{b} \succeq \mathbf{0},\, \mathbf{1}^T\mathbf{b} = 1\}$ and the last step is used to smooth the objective function via introducing one more auxiliary optimization variable $\mathbf{b}$ [36]. Then, we have the following conclusion on the Karush-Kuhn-Tucker solution to (35), which also holds for $\mathcal{Q}_1$.

Lemma 1. *Let*

$$U(\mathbf{v}') = \frac{\sqrt{\beta}\mathbf{C}(\mathbf{v}')\arg\min_{\mathbf{b}\in\Delta}\Phi\left(\mathbf{v}',\mathbf{b}\right)}{\|\mathbf{C}(\mathbf{v}')\arg\min_{\mathbf{b}\in\Delta}\Phi\left(\mathbf{v}',\mathbf{b}\right)\|_2}, \tag{36}$$

where

$$\Phi\left(\mathbf{v}',\mathbf{b}\right) = 2\sqrt{\beta}\|\mathbf{C}(\mathbf{v}')\mathbf{b}\|_2 - \left[\mathbf{q}(\mathbf{v}')\right]^T\mathbf{b}, \tag{37}$$

$$\mathbf{C}(\mathbf{v}') = \left[\frac{\mathbf{g}_1\mathbf{g}_1^H\mathbf{v}'}{\sigma_1^2},\cdots,\frac{\mathbf{g}_K\mathbf{g}_K^H\mathbf{v}'}{\sigma_K^2}\right] \in \mathbb{C}^{N\times K}, \tag{38}$$

$$\mathbf{q}(\mathbf{v}') = \left[\frac{|\mathbf{g}_1^H\mathbf{v}'|^2}{\sigma_1^2},\cdots,\frac{|\mathbf{g}_K^H\mathbf{v}'|^2}{\sigma_K^2}\right]^T \in \mathbb{C}^{K\times 1}. \tag{39}$$

Then with any feasible $\mathbf{v}^{(0)}$ *and fixed point iteration* $\mathbf{v}^{(n+1)} \leftarrow U(\mathbf{v}^{(n)})$*, every limit point* $\mathbf{v}^{\diamond}$ *of the sequence* $\{\mathbf{v}^{(0)},\mathbf{v}^{(1)},...\}$ *is a Karush-Kuhn-Tucker solution to problem* (35)*.*

Proof. It can be proved that $U(\mathbf{v}')$ is the optimal solution to $\min_{\|\mathbf{v}\|_2^2\le\beta}\max_{\mathbf{b}\in\Delta} g(\mathbf{v},\mathbf{v}',\mathbf{b})$, where $g(\mathbf{v},\mathbf{v}',\mathbf{b})$ is a surrogate function satisfying $g(\mathbf{v},\mathbf{v}',\mathbf{b}) \ge h(\mathbf{v},\mathbf{b})$, $g(\mathbf{v}',\mathbf{v}',\mathbf{b}) = h(\mathbf{v}',\mathbf{b})$, and $\nabla g(\mathbf{v}',\mathbf{v}',\mathbf{b}) \ge \nabla h(\mathbf{v}',\mathbf{b})$. According to [35] and the properties of $g(\mathbf{v},\mathbf{v}',\mathbf{b})$, every limit point of the sequence $(\mathbf{v}^{(0)},\mathbf{v}^{(1)},\cdots)$ generated by $\mathbf{v}^{(n+1)} \leftarrow U(\mathbf{v}^{(n)})$ and a feasible $\mathbf{v}^{(0)}$ is the KKT solution to (35). More details can be found in Appendix C. ■

Although Lemma 1 reveals the solution structure of (35), computation of $U(\mathbf{v}')$ is not straightforward as it involves another optimization problem of $\mathbf{b}$, which should also be solved with low computation costs. In the following, we will solve the optimization problem of $\mathbf{b}$ via the methods of smoothing and acceleration.

B. Optimization of $\mathbf{b}$ *via Smoothing and Acceleration*

In order to compute $U(\mathbf{v}')$, a necessary step is to find the optimal vector

$$\mathbf{b}^* = \arg\min_{\mathbf{b}\in\Delta}\Phi\left(\mathbf{b}\right), \tag{40}$$

where we have omitted the symbol $\mathbf{v}'$ in (37) since $\mathbf{v}'$ is a known and fixed vector in each iteration. Notice that the gradient of the objective

$$\nabla_{\mathbf{b}}\Phi(\mathbf{b}) = \frac{2\sqrt{\beta}\,\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right)}{\|\mathbf{C}\mathbf{b}\|_2} - \mathbf{q} \tag{41}$$

is unbounded when $\|\mathbf{Cb}\|_2 \to 0$, which happens if $\mathbf{b} \in \mathrm{Null}(\mathbf{C})$. Therefore, it is nontrivial to apply first-order method to problem (40). To avoid the unbounded gradients, we adopt the smoothing technique [37] to replace $\Phi(\mathbf{b})$ in (40) with

$$\Xi(\mathbf{b}) = 2\sqrt{\beta} \times \sqrt{\phi^2 + \left|\left|\mathbf{Cb}\right|\right|_2^2} - \mathbf{q}^T\mathbf{b}, \tag{42}$$

where the tuning parameter $\phi \geq 0$ such that $\Xi(\mathbf{b}) = \Phi(\mathbf{b})$ for $\phi = 0$. Then problem (40) can be approximated by

$$\mathcal{Q}_2 : \min_{\mathbf{b}\in\Delta} \ \Xi\left(\mathbf{b}\right). \tag{43}$$

In the following, we first elaborate the optimal solution of problem $\mathcal{Q}_2$, and then establish the relation between the solutions of problems $\mathcal{Q}_2$ and (40). First of all, we have the following lemma on the objective of problem $\mathcal{Q}_2$.

Lemma 2. *$\Xi(\mathbf{b})$ is Lipschitz smooth for $\mathbf{b} \in \Delta$, with the Lipschitz constant of gradient*

$$L_\Xi(\phi) = \frac{2\sqrt{\beta}\,\lambda_{\max}\left[\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right)\right]}{\sqrt{\phi^2 + \lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right)/K}}. \tag{44}$$

Proof. The proof is based on bounding the eigenvalue of the Hessian matrix of $\Xi(\mathbf{b})$. More details can be found in Appendix D. ■

Lemma 2 shows that $\mathcal{Q}_2$ is a Lipschitz smooth problem. As a result, the acceleration method [38], [39] can be adopted to optimally solve $\mathcal{Q}_2$ iteratively. The algorithm is summarized in Theorem 3.

Theorem 3. *Let $\mathbf{b}(0) \in \Delta$ and*

$$\mathbf{b}(m+1) = \Pi_\Delta\Big[\boldsymbol{\rho}(m) - \frac{1}{L_\Xi(\phi)}\nabla_{\mathbf{b}}\Xi(\mathbf{b})\Big|_{\mathbf{b}=\boldsymbol{\rho}(m)}\Big], \tag{45}$$

where m is the iteration index, Π_Δ is the projection onto set Δ, $L_\Xi(\phi)$ is defined in (44)*, and*

$$\nabla_{\mathbf{b}}\Xi(\mathbf{b}) = \mathbf{q} - 2\sqrt{\beta} \times \frac{\mathrm{Re}\left(\mathbf{C}^H\mathbf{Cb}\right)}{\sqrt{\phi^2 + \|\mathbf{Cb}\|_2^2}}, \tag{46}$$

$$\boldsymbol{\rho}(m) = \mathbf{b}(m) + \frac{c(m-1)-1}{c(m)}\left(\mathbf{b}(m) - \mathbf{b}(m-1)\right), \tag{47}$$

$$c(m) = \frac{1}{2}\left(1 + \sqrt{1 + 4\left(c(m-1)\right)^2}\right), \quad c(0) = 1. \tag{48}$$

Then the sequence computed from (45)–(48) *converges to the optimal solution of $\mathcal{Q}_2$ with an iteration complexity $\mathcal{O}\left(\sqrt{L_\Xi(\phi)/\epsilon}\right)$, where ϵ is the target accuracy.*

Proof. It can be proved by following a similar approach in [39, Theorem 4.4]. ■

Notice that the iteration complexity touches the lower bound derived in [40, Theorem 2.1.6]. The computation of the projection $\Pi_{\Delta}(\mathbf{u})$ given the input vector $\mathbf{u}$ is summarized in Lemma 3.

Lemma 3. *Let* $\mathbf{u}' = \mathrm{sort}(\mathbf{u})$*, where the function* sort *permutes the elements of* $\mathbf{u}$ *in a descent order such that* $u'_1 \geq \cdots \geq u'_K$ *and* $\delta = \max_{x\in\{1,\cdots,K\}} \left\{x : \frac{\sum_{l=1}^{x} u'_l - 1}{x} < u'_x\right\}$*. Then*

$$\Pi_{\Delta}(\mathbf{u}) = \left(\mathbf{u} - \frac{\sum_{l=1}^{\delta} u'_l - 1}{\delta}\right)^{+}. \tag{49}$$

Proof. Please refer to [41, Proposition 2.2]. ■

Finally, we have the following conclusion on the relation between the solutions to problems $\mathcal{Q}_2$ and (40).

Theorem 4. *(i) If* $\mathrm{Rank}\left([\mathbf{g}_1, \cdots, \mathbf{g}_K]\right) = K$ *(thus* $L_{\Xi}(0) < +\infty$*), the optimal solution to problem* $\mathcal{Q}_2$ *is optimal to problem* (40) *by setting* $\phi = 0$*. (ii) If* $\mathrm{Rank}\left([\mathbf{g}_1, \cdots, \mathbf{g}_K]\right) \neq K$*, then* $L_{\Xi}(0) = +\infty$*, and* $L_{\Xi}(\phi) < +\infty$ *if* $\phi > 0$*. (iii) For all* $\mathbf{b}' \in \Delta$ *with* $\Xi(\mathbf{b}') - \Xi(\mathbf{b}^{\diamond}) \leq \epsilon$*,*

$$\Phi(\mathbf{b}') - \Phi(\mathbf{b}^{*}) \leq 2\sqrt{\beta}\,\phi + \epsilon, \tag{50}$$

where $\mathbf{b}^{\diamond}$ *and* $\mathbf{b}^{*}$ *denote the optimal solutions to* $\mathcal{Q}_2$ *and* (40)*, respectively.*

Proof. Please refer to Appendix E. ■

Part (i) of Theorem 4 indicates that we can always set $\phi = 0$ if the user channels are independent. In this case, $\Phi(\mathbf{b}) = \Xi(\mathbf{b})$, which means that the optimal solution to problem $\mathcal{Q}_2$ is the same as that of (40). On the other hand, part (ii) of Theorem 4 indicates that if the user channels are correlated, we must choose $\phi > 0$, and the conversion from (40) to $\mathcal{Q}_2$ would lead to approximation error. However, this error is controllable by choosing a small $2\sqrt{\beta}\,\phi$ according to part (iii) of Theorem 4 (e.g., with $\phi = 0.1/(2\sqrt{\beta})$, the approximation error is at most $2\sqrt{\beta}\,\phi = 0.1$).

C. Summary and Complexity Analysis of AGP

For the proposed AGP algorithm, the accelerated random coordinate descent is first used to compute $\mathbf{w}^{\diamond}$ for problem on the right hand side of the constraint in (33b), which requires a complexity of $\mathcal{O}\left(KN\right)$. To optimize $\mathbf{v}^{\diamond}$, in each fixed-point iteration, the terms $\mathbf{C}$ in (38) and

$\mathbf{q}$ in (39) are computed with a complexity of $\mathcal{O}(KN)$, followed by the iterative calculation of variable $\mathbf{b}$ in (40) with equations (45)–(48), which involves a complexity of $\mathcal{O}(KN)$ for gradient computation. Therefore, the overall complexity of AGP for solving $\mathcal{Q}$ is $\mathcal{O}(KN)$. Notice that with the obtained $\mathbf{w}^{\diamond}$ and $\mathbf{v}^{\diamond}$, we need to recover $\{\mathbf{F}^{\star}, r_k^{\star}, t_k^{\star}\}$. To satisfy the unit-modulus constraints, $\mathbf{w}^{\diamond}$ and $\mathbf{v}^{\diamond}$ are projected onto the unit-modulus constraints as $\mathbf{w}^{\star} = \exp(\mathrm{j}\angle\mathbf{w}^{\diamond})$ and $\mathbf{v}^{\star} = \exp(\mathrm{j}\angle\mathbf{v}^{\diamond})$. The recovered phase shift network is $\mathbf{F}^{\star} = (\mathbf{w}^{\star})^H\mathbf{v}^{\star}$. With $\mathbf{F}^{\star}$, $t_k^{\star} = 1/(K(\mathbf{w}^{\star})^H\mathbf{h}_k)$ and $r_k^{\star}$ is computed using (27).

VI. Simulation Results and Discussions

This section presents simulation results to verify the performance of the proposed scheme. The pathloss of the user k is set to $\varrho_k = -60\,\mathrm{dB}$, and $\mathbf{h}_k$ and $\mathbf{g}_k$ are generated according to $\mathcal{CN}(\mathbf{0}, \varrho_k\mathbf{I}_N)$. It is assumed that the channels during the model uploading and aggregation procedure are static in each federated learning iteration. On the other hand, different iterations adopt independent channels and noise realizations. The power scaling factor $\gamma = 1$ at the server. All problems are solved by Matlab R2019a on a desktop with an Intel Core i7-7700 CPU at 3.6 GHz and 16 GB RAM. The Interior point method is implemented using CVX Mosek [32]. Under the above setting, we consider the following benchmark schemes for comparison.

1) The optimized user transceiver scheme, which can be viewed as the extension of [14], [15] to the multi-antenna scenarios. This scheme optimizes transceiver designs $\{t_k, r_k\}$ but adopts unoptimized beamformer $\mathbf{F} = \mathbf{I}_N$. The transmitters $\{t_k\}$ and receivers $\{r_k\}$ are optimized iteratively by solving (17c) using CVX and (17b) using (27), respectively.
2) The digital beamforming scheme in [16], which ignores the unit-modulus constraints on $\mathbf{F}$. As such, the obtained MSE (training loss, test error) of the digital beamfomring scheme serves as a lower bound on that of the proposed scheme.
3) The digital beamformer projection scheme, which sets $\mathbf{F}$ as the projection of the digital beamforming design in [17] onto the unit-modulus constraints.
4) The UMAirComp scheme with SDR and CVX [32]. The analog beamformer $\mathbf{F}$, the transmitters $\{t_k\}$, and the receivers $\{r_k\}$ are optimized iteratively by solving (17a) using SDR, (17c) using CVX, and (17b) using (27), respectively.[5]

[5]If the matrix solution to the SDR problem of (17a) is not rank-one, we use the principal component of the obtained matrix as the phase shift design.

A. Performance Evaluation of PAM-based and AGP-based UMAirComp

First, to verify the learning performance of PAM-based UMAirComp, we consider the image classification task based on a convolutional neural network (CNN). The mixed national institute of standards and technology (MNIST) dataset [42] is used, where $\mathbf{d}_{k,l}^{\text{in}} \in \mathbb{R}^{784\times 1}$ is a gray-scale vector of images and $\mathbf{d}_{k,l}^{\text{out}} \in \mathbb{R}^{10\times 1}$ is a label vector containing only one non-zero element. The CNN consists of two convolution layers, two max pooling layers, and a fully-connected layer. The loss function

$$\Theta(\mathbf{d}_{k,l}, \mathbf{x}_k) = \left[f_{\text{cnn}}\left(\mathbf{x}_k, \mathbf{d}_{k,l}^{\text{in}}\right) - \mathbf{d}_{k,l}^{\text{out}}\right]^2 /2, \tag{51}$$

where $f_{\text{cnn}}\left(\mathbf{x}_k, \mathbf{d}_{k,l|}^{\text{in}}\right)$ is the softmax output of CNN. The training step-size is $\varepsilon = 1$, the number of local epochs is 4, and the total number of FL iterations is set to 400. For each iteration, the maximum transmit powers at users are $P_0 = 10\ \text{mW}$ (i.e., $10\,\text{dBm}$). The noise powers at the server and users are set as $-80\,\text{dBm}$, which capture the effects of thermal noise, receiver noise, and interference. We set $M_{\max} = Q_{\max} = 200$ and $N_{\max} = 20$ for the inner and outer layer iterations for the PAM algorithm.

Comparison among four benchmark schemes and the proposed scheme is shown in Fig. 4. Specifically, the axis of the radar map in Fig. 4a ranges from 0 to 0.3 for MSE (i.e., the objective function of $\mathcal{P}$), training loss, and test error. The MSE is obtained by averaging over the 400 FL iterations and the training loss (test error) is obtained when the entire FL procedure terminates. Since our goal is to minimize all these metrics concurrently, a smaller area indicates better performance. Firstly, the optimized user transceiver scheme has the largest area, which can be treated as a worst-case performance bound. This is because the phase shift network at the server does not align with the users' channels with the optimized user transceiver scheme. Secondly, the proposed PAM-based UMAirComp scheme achieves a similar size of triangle as that of the digital beamforming scheme. This is because for federated learning systems, there is no need to decode all the local model parameters and the beam direction can be less accurate. As such, analog beamforming leads to negligible performance loss compared with the digital beamforming. Furthermore, the area of the proposed scheme is completely covered by that of the digital beamformer projection scheme and the area reduction of the former comes from the analog beamforming design based on PAM. Finally, the scheme with SDR and CVX also achieves competing performance. However, it is at the cost of high computational complexities and execution time, making it not applicable in practice.

On the other hand, Fig. 4b and Fig. 4c show the training loss and the test error versus the number of FL iterations, respectively. Firstly, due to the high MSE (close to 0.1), the optimized user transceiver scheme diverges. Secondly, the FL scheme with digital beamformer projection is competitive compared with the proposed scheme at the beginning, but the training loss does not further decrease with the increasing number of FL iterations. Indeed, the test error increases after 200 iterations. This means that model parameter errors have a stronger impact as the number of FL iterations increases. This is because deep learning models are usually over parameterized. Thus, as the training procedure gets closer to convergence, the model parameters would become sparser, which are more sensitive to errors. Lastly, the proposed scheme achieves a training loss curve and a test error curve close to their associated lower bounds. This again demonstrates the small performance gap between the analog and the digital beamforming in federated learning systems.

To evaluate the solution quality and running time of the proposed AGP-based UMAirComp, we simulate the case of $N = \{8, 16, 32, 128\}$ with $K = 4$. It can be seen from Fig. 5a that the scheme with SDR and CVX is the most time consuming, and it fails in providing a solution within a reasonable amount of time for the case of $N = \{32, 128\}$. The proposed AGP-based UMAirComp requires running time two orders of magnitude smaller than that of all the other schemes. Note that the lower bound is obtained from the digital beamforming scheme, which tackles a less challenging problem without unit-modulus constraints. The AGP algorithm achieves MSE performance close to that of the proposed PAM algorithm and the lower bound as shown in Fig. 5b. This indicates that the adopted approximations in the AGP algorithm lead to small performance loss in practice.

B. UMAirComp for V2X Autonomous Driving

Next, to verify the robustness of the proposed UMAirComp framework in complex learning tasks, we consider the object detection task for V2X autonomous driving [43]. In particular, we propose a V2X autonomous driving simulation platform shown in Fig. 5c, which is a virtual-reality system with close-to-reality environments, interactions, and interfaces. The platform is developed based on Car Learning to Act (CARLA) [27], Simulation of Urban MObility (SUMO), Robot Operating System (ROS), and Pytorch in Ubuntu 18.04 with a GeForce GTX 1080 GPU. The platform consists of learning, communication, and algorithm modules, where the learning and communication modules interact with each other by exchanging model parameters while

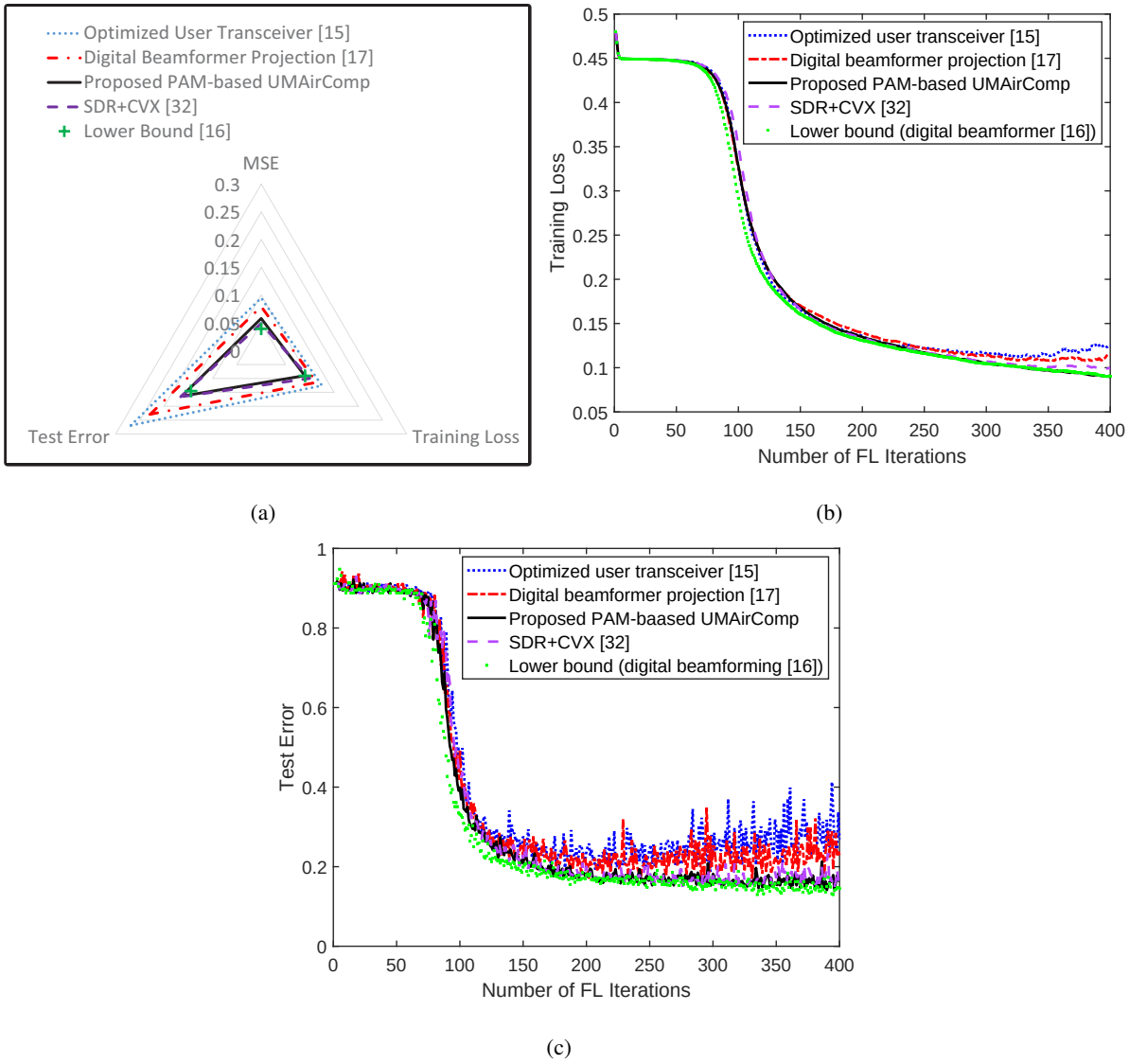
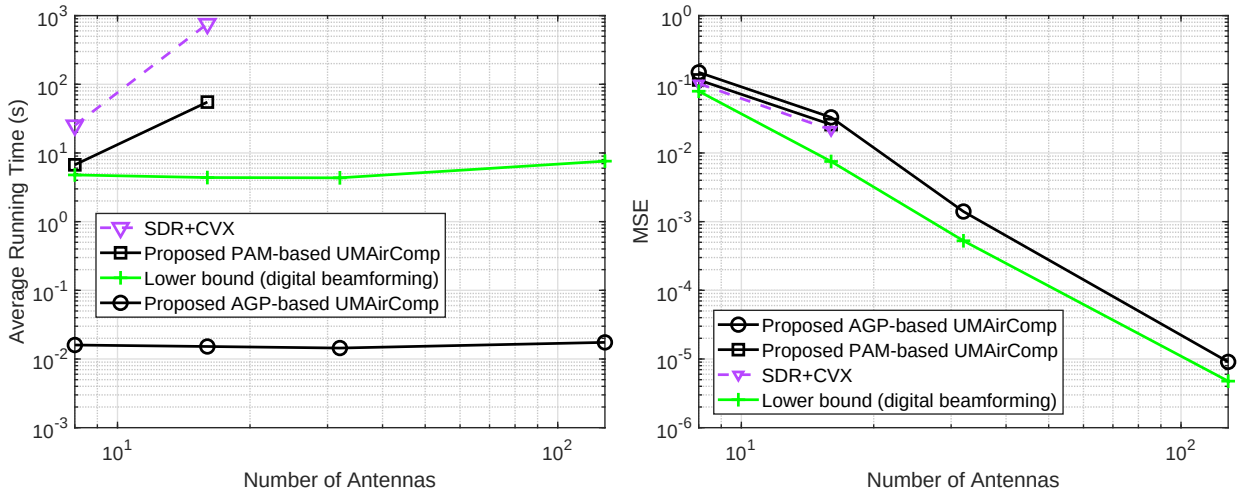


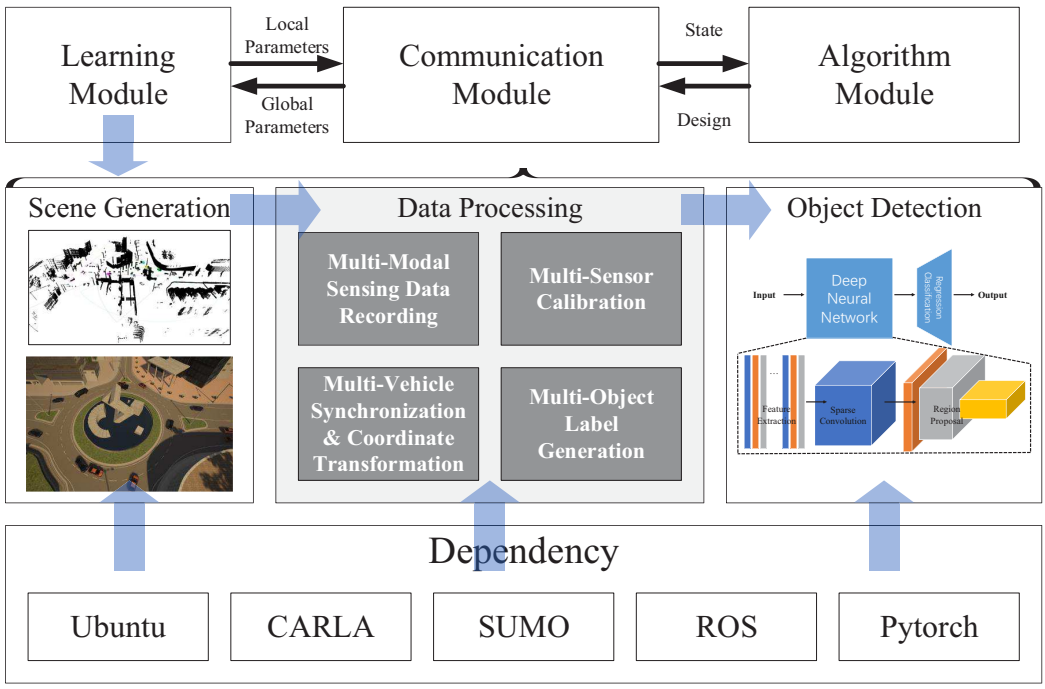
Fig. 4: Comparison between the proposed and benchmark schemes when $N = 8$ with $K = 10$: a) comparison of normalized MSE, training loss, and test error; b) training loss versus the number of FL iterations; c) worst test error among all users versus the number of FL iterations.

the algorithm module controls the communication module by exchanging system/channel state information and optimized designs.

The learning module is a function cluster consisting of scenario generation, data processing, and object detection. We use CARLA, which is a client-server platform based on Unreal Engine, to construct a high-fidelity scenario. The "Town02" map is generated with 28 vehicles, including 4 autonomous vehicles. The entire scenario lasts for 90 seconds and contains 700 frames at each



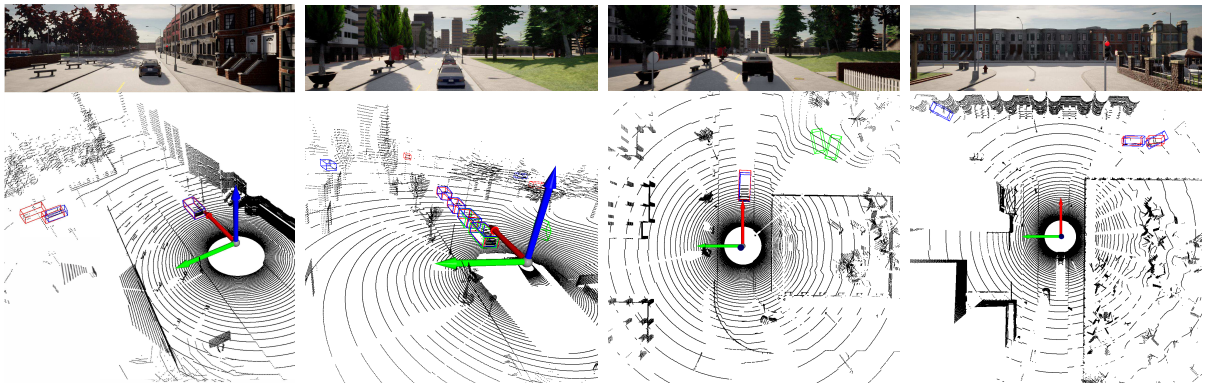
(a) (b)



(c)

Fig. 5: a) Average running time versus the number of antennas when $K = 4$; b) normalized MSE versus the number of antennas when $K = 4$; c) Architecture of the proposed V2X autonomous driving platform based on CARLA.

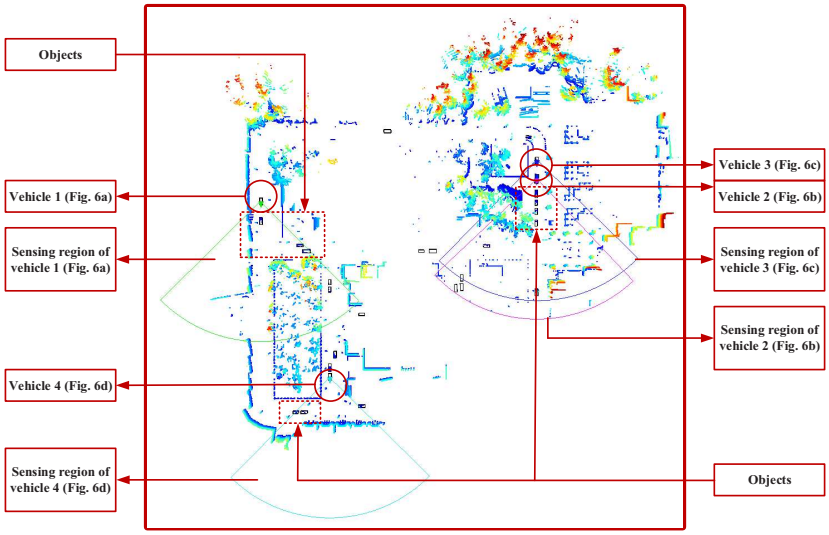
autonomous vehicle. The first 200 frames are used for training at each vehicle. The latter 500 frames are used for inference and testing. On the other hand, data processing involves the following steps.



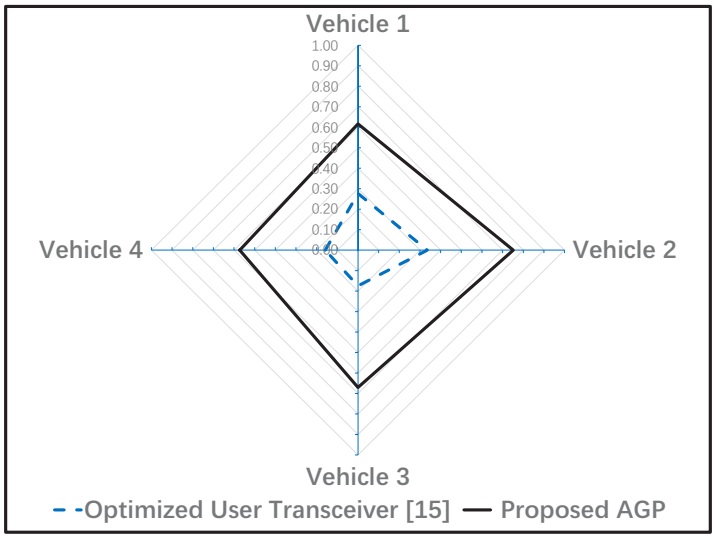
(a) (b) (c) (d)

Fig. 6: Detection results when $N = 500$ with $K = 4$. The red box is the ground truth; the blue box is from the proposed AGP-based UMAirComp scheme; the green box is from the optimized user transceiver scheme (benchmark) scheme. a) the benchmark scheme detects nothing while the proposed scheme detects two objects; b) the benchmark scheme only detects two nearby objects while the proposed scheme can detect far-away objects; c) the benchmark scheme generates false positive results while the proposed scheme generates accurate prediction; d) the benchmark scheme cannot detects occlusion objects while the proposed scheme detects all of them.

1) **Multi-modal sensory data recording**. Recording sensory data is implemented based on Python application programming interfaces (APIs) at the CARLA client [27]. Each autonomous vehicle (Tesla Model 3) is equipped with a GPS, a camera, and a 64-line LiDAR at 10 Hz. The default LiDAR range is set to 100 m, and its FoV is 90° in the front. Each frame of point cloud is considered as an input data sample $\mathbf{d}_{k,l}^{\text{in}}$. All the above information is stored in a sensor file.
2) **Multi-sensor calibration**. Since each vehicle has three coordinate systems (i.e., LiDAR coordinate, camera coordinate, and GPS coordinate), we need to compute the rotation and transition matrices among these coordinate systems [44]. Furthermore, since we have multiple cameras, we need to compute the ration matrices from the reference camera to other cameras. Lastly, projection matrices are required to associate a point in the 3D space with a point in the camera image. All the above information is stored in a calibration file.
3) **Multi-vehicle synchronization and coordinate transformation**. All vehicles should be synchronized [45]. We adopt the synchronization framework provided by CARLA-SUMO



(a)



(b)

Fig. 7: a) The global bird eye view of the frame in Fig. 7. The green vehicle is in Fig. 6a. The pink vehicle is in Fig. 6b. The blue vehicle is in Fig. 6c. The cyan vehicle is in Fig. 6d. The figure demonstrates the heterogeneity of the distributed datasets at different vehicles. b) Comparison between the proposed AGP-based UMAirComp and the optimized user transceiver schemes when $N = 500$ with $K = 4$.

integration to synchronize the sensors on each vehicle. We exploit the time stamps of the 3D laser scanner as a reference and consider each spin as a frame. On the other hand, different vehicles have different local coordinate systems. As such, we use the APIs provided by the

CARLA worlds, maps, and actors to transform these local coordinates to global coordinates. All the above information is stored in a global coordinate file.

4) **Multi-object label generation**. The ground truth labels $\{\mathbf{d}_{k,l}^{\text{out}}\}$ for object detection should satisfy the KITTI format. Each label $\mathbf{d}_{k,l}^{\text{out}}$ consists of 16 elements [44]: 1) category; 2) boundary or not; 3) occlusion or not; 4) observation angle; 5–6) left top pixel coordinates in the camera image; 7–8) right bottom pixel coordinates in the camera image; 9) height of object; 10) width of object; 11) length of object; 12-14) 3D object location in the camera coordinate system; 15) rotation of object; 16) detection confidence. All the above information is stored in a label file.[6]

Finally, the sparsely embedded convolutional detection (SECOND) neural network [25] is used for object detection on the processed dataset. The local model structure can be found in [25]. The loss function

$$\Theta(\mathbf{d}_{k,l}, \mathbf{x}_k) = f_{\text{class}}\left(\mathbf{x}_k, \mathbf{d}_{k,l}\right) + f_{\text{box}}\left(\mathbf{x}_k, \mathbf{d}_{k,l}\right) + f_{\text{orientation}}\left(\mathbf{x}_k, \mathbf{d}_{k,l}\right), \tag{52}$$

where $\mathbf{d}_{k,l} = \left(\mathbf{d}_{k,l}^{\text{in}}, \mathbf{d}_{k,l}^{\text{out}}\right)$, f_{class} is the classification loss, f_{box} is the box regression loss, and $f_{\text{orientation}}$ is the softmax orientation estimation loss. The SECOND network is trained with a diminishing learning rate, where the initial learning rate is set to 10^{-4} and the number of local updates is $E = 1$. The average precision at intersection over union IoU$= 0.5$ is used for performance evaluation.

For the communication module, the case of $N = 500$ and $K = 4$ is simulated. It is assumed that the sensing datasets are generated and pre-stored at the vehicles before the FL procedure. The total number of FL iterations is 8, with independently generated channel in each FL iteration. The maximum transmit power of each autonomous vehicle is set to $P_0 = 1\,\text{W}$ (i.e., $30\ \text{dBm}$), and the noise powers are set as $-90\,\text{dBm}$.

The detection results and the global bird eye view of a particular frame are shown in Fig. 6 and Fig. 7a, respectively. The comparison between the proposed scheme and the optimized user transceiver scheme [15] is provided in Fig. 7b. From the above results, it can be seen

[6]In practice, to obtain the training labels of objects, each vehicle broadcasts its ego position and waits for messages from the nearby infrastructures. Since the infrastructures are fixed at utility poles and connected to servers via wirelines, they have broader fields of views (FoVs) and deeper neural networks than those of vehicles. Thus, their outputs are more accurate, which can be transmitted to vehicles via the vehicle-to-infrastructure (V2I) communication and adopted as pseudo labels (i.e., ground-truth labels with noises).

that the test errors of both schemes in Fig. 7b are worse than their counterparts in Fig. 4a. In contrast, the MSEs of both schemes are remarkably better than their counterparts in Fig. 4a.[7] This implies that object detection tasks in autonomous driving are more sensitive to model parameter errors. This is because the trained model parameters for the SECOND network are generally sparser and a slight model error would lead to completely different detections. Furthermore, as seen from the objective of $\mathcal{P}$, the model error of user k with the UMAirComp framework is dominated by $\gamma \sum_{j=1}^{K} \left| r_k \mathbf{g}_k^H \mathbf{F} \mathbf{h}_j t_j - \alpha_j \right|^2$. Therefore, the methods to minimize the model errors can be categorized into 1) configuration of wireless channels $\{\mathbf{h}_k, \mathbf{g}_k\}$ and 2) design of analog beamformer $\mathbf{F}$. For the configuration of wireless channels, due to $\alpha_1 = \cdots = \alpha_K$, the ideal wireless channels for the proposed UMAirComp satisfy $\mathbf{g}_1 = \cdots = \mathbf{g}_K$ and $\mathbf{h}_1 = \cdots = \mathbf{h}_K$. This implies that the UMAirComp should be executed when vehicles are geographically close to each other such that the magnitudes of $\{\mathbf{h}_k, \mathbf{g}_k\}$ are close. This is the case in vehicle platooning and vehicle parking scenarios [45]; otherwise, some emerging techniques (e.g., reconfigurable intelligence surfaces (RIS) [5]) should be adopted to smartly alter the wireless environment. On the other hand, for the design of beamformers, the key is to align the various channels $\{\mathbf{h}_k, \mathbf{g}_k\}$ to a same direction and power for decoding superimposed signals. This is the case of Fig. 6b, where the proposed method achieves much larger average precisions than the optimized user transceiver scheme for all vehicles. Note that the running time of AGP-based UMAirComp is only 0.06 s, which can be further accelerated via GPU.

VII. Conclusion

This paper proposed the UMAirComp framework to support simultaneous transmission of local model parameters in edge federated learning systems. Training loss upper bounds of UMAirComp were derived, which reveal that the key to minimize FL training loss is to minimize the maximum MSE among all users. Two low-complexity large-scale optimization algorithms were proposed to tackle the nonconvex nonsmooth loss bound minimization problem. The performance and runtime of the UMAirComp framework with the proposed optimization algorithms were verified using the image classification task. The performance of the proposed framework and algorithms were also verified in a V2X autonomous driving simulation platform and experimental results

[7]The normalized MSEs of the proposed AGP-based UMAirComp and the optimized user transceiver (benchmark) schemes in Fig. 7b are only 6×10^{-11} and 0.0013, respectively. The normalized MSEs of the two schemes in Fig. 4a are larger than 0.05.

have shown that the object detection precision with the proposed algorithm is significantly higher than that achieved by benchmark schemes.

APPENDIX A
PROOF OF THEOREM 1

First, let $\Delta\mathbf{x}^{[i]} = \mathbf{x}_k^{[i+1]}(0) - \left[\mathbf{x}_k^{[i]}(0) - \varepsilon\nabla_{\mathbf{x}}\Lambda\left(\mathbf{x}_k^{[i]}(0)\right)\right]$ and $\|\Delta\mathbf{x}^{[i]}\|_2^2$ can be upper bounded by

$$\begin{aligned}
\|\Delta\mathbf{x}^{[i]}\|_2^2 &= \left\|\mathbf{x}_k^{[i+1]}(0) - \mathbf{x}_k^{[i]}(0) + \varepsilon\nabla_{\mathbf{x}}\Lambda\left(\mathbf{x}_k^{[i]}(0)\right)\right\|_2^2 \\
&= \Bigg\|\mathbf{x}_k^{[i+1]}(0) - \boldsymbol{\theta}^{[i]} + \sum_{j=1}^{K}\alpha_j\mathbf{x}_j^{[i]}(0) - \varepsilon\sum_{j=1}^{K}\frac{\alpha_j}{|\mathcal{D}_j|}\sum_{\mathbf{d}_{j,l}\in\mathcal{D}_j}\nabla_{\mathbf{x}}\Theta\left(\mathbf{d}_{j,l},\mathbf{x}_j^{[i]}(0)\right) \\
&\quad - \mathbf{x}_k^{[i]}(0) + \varepsilon\nabla_{\mathbf{x}}\Lambda(\mathbf{x}_k^{[i]}(0))\Bigg\|_2^2 \\
&\le \left\|\mathbf{x}_k^{[i+1]}(0) - \boldsymbol{\theta}^{[i]}\right\|_2^2 + \sum_{j=1}^{K}\alpha_j\left\|\mathbf{x}_j^{[i]}(0) - \mathbf{x}_k^{[i]}(0)\right\|_2^2 \\
&\quad + \varepsilon^2\sum_{j=1}^{K}\left\|\frac{\alpha_j}{|\mathcal{D}_j|}\left(\sum_{\mathbf{d}_{j,l}\in\mathcal{D}_j}\nabla_{\mathbf{x}}\Theta\left(\mathbf{d}_{j,l},\mathbf{x}_j^{[i]}(0)\right) - \sum_{\mathbf{d}_{j,l}\in\mathcal{D}_j}\nabla_{\mathbf{x}}\Theta\left(\mathbf{d}_{j,l},\mathbf{x}_k^{[i]}(0)\right)\right)\right\|_2^2, \quad (53)
\end{aligned}$$

where the second equality is obtained from (2) with $E = 1$ and (10), and the inequality is obtained due to $\|\mathbf{a}_1 + \mathbf{a}_2\|_2^2 \le \|\mathbf{a}_1\|_2^2 + \|\mathbf{a}_2\|_2^2$.

On the other hand, according to (9), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{x}_k^{[i+1]}(0) - \boldsymbol{\theta}^{[i]}\right\|_2^2\right] = \mathbb{MSE}_k^{[i]}, \\
&\mathbb{E}\left[\left\|\mathbf{x}_k^{[i]}(0) - \mathbf{x}_j^{[i]}(0)\right\|_2^2\right] = \mathbb{E}\left[\left\|\mathbf{x}_k^{[i]}(0) - \boldsymbol{\theta}^{[i-1]} + \boldsymbol{\theta}^{[i-1]} - \mathbf{x}_j^{[i]}(0)\right\|_2^2\right] \le 2\mathbb{MSE}_k^{[i]}. \quad (54)
\end{aligned}$$

Moreover, according to Assumption 1, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{|\mathcal{D}_j|}\left(\sum_{\mathbf{d}_{j,l}\in\mathcal{D}_j}\nabla_{\mathbf{x}}\Theta\left(\mathbf{d}_{j,l},\mathbf{x}_j^{[i]}(0)\right) - \sum_{\mathbf{d}_{j,l}\in\mathcal{D}_j}\nabla_{\mathbf{x}}\Theta\left(\mathbf{d}_{j,l},\mathbf{x}_k^{[i]}(0)\right)\right)\right\|_2^2\right] \\
&\le \mathbb{E}\left[L^2\|\mathbf{x}_k^{[i]}(0) - \mathbf{x}_j^{[i]}(0)\|_2^2\right] \le 2L^2\,\mathbb{MSE}_k^{[i]}. \quad (55)
\end{aligned}$$

Putting (54) and (55) into (53), and according to the expression of ε yields

$$\mathbb{E}\left[\|\Delta\mathbf{x}^{[i]}\|_2^2\right] \le \left(3 + 2\sum_{j=1}^{K}\alpha_j^2\right)\max_{k=1,\cdots,K}\mathbb{MSE}_k^{[i]}. \quad (56)$$

Due to $\frac{1}{|\mathcal{D}_k|}\sum_{\mathbf{d}_{k,l}\in\mathcal{D}_k}\nabla_{\mathbf{x}}^2\Theta(\mathbf{d}_{k,l},\mathbf{x})\preceq L\mathbf{I}$, we have $\nabla_{\mathbf{x}}^2\Lambda(\mathbf{x})\preceq 1/L\mathbf{I}$. Based on $\mu\mathbf{I}\preceq\nabla_{\mathbf{x}}^2\Lambda(\mathbf{x})\preceq 1/L\mathbf{I}$, the following equations hold [40]:

$$\Lambda(\mathbf{x}')\leq\Lambda(\mathbf{x})+(\mathbf{x}'-\mathbf{x})^T\nabla\Lambda(\mathbf{x})+\frac{L}{2}\|\mathbf{x}'-\mathbf{x}\|_2^2, \tag{57a}$$

$$\Lambda(\mathbf{x}')\geq\Lambda(\mathbf{x})+(\mathbf{x}'-\mathbf{x})^T\nabla\Lambda(\mathbf{x})+\frac{\mu}{2}\|\mathbf{x}'-\mathbf{x}\|_2^2. \tag{57b}$$

Putting $\mathbf{x}'=\mathbf{x}_k^{[i+1]}(0)=\mathbf{x}_k^{[i]}(0)-\varepsilon\nabla_{\mathbf{x}}\Lambda\left(\mathbf{x}_k^{[i]}(0)\right)+\Delta\mathbf{x}^{[i]}$ and $\mathbf{x}=\mathbf{x}_k^{[i]}(0)$ into (57a), we have

$$\Lambda\left(\mathbf{x}_k^{[i+1]}(0)\right)\leq\Lambda\left(\mathbf{x}_k^{[i]}(0)\right)-\frac{1}{2L}\left\|\nabla_{\mathbf{x}}\Lambda\left(\mathbf{x}_k^{[i]}(0)\right)\right\|_2^2+\frac{L}{2}\|\Delta\mathbf{x}^{[i]}\|_2^2. \tag{58}$$

On the other hand, the right hand side of (57b) is minimized at $\mathbf{x}'=\mathbf{x}-\mu^{-1}\nabla_{\mathbf{x}}\Lambda(\mathbf{x})$. Putting this expression and $\mathbf{x}=\mathbf{x}_k^{[i]}(0)$ into (57b) gives

$$\|\nabla_{\mathbf{x}}\Lambda(\mathbf{x}_k^{[i]}(0))\|_2^2\geq 2\mu\left[\Lambda(\mathbf{x}_k^{[i]}(0))-\Lambda(\boldsymbol{\theta}^*)\right]. \tag{59}$$

Combining (58) and (59) gives

$$\Lambda(\mathbf{x}_k^{[i+1]}(0))-\Lambda(\mathbf{x}_k^{[i]}(0))\leq\left(1-\frac{\mu}{L}\right)\left[\Lambda(\mathbf{x}_k^{[i]}(0))-\Lambda(\boldsymbol{\theta}^*)\right]+\frac{L}{2}\|\Delta\mathbf{x}^{[i]}\|_2^2. \tag{60}$$

Applying this recursively leads to

$$\begin{aligned}\Lambda(\mathbf{x}_k^{[i+1]}(0))-\Lambda(\boldsymbol{\theta}^*)\leq&\left(1-\frac{\mu}{L}\right)^{i+1}\left[\Lambda(\mathbf{x}_k^{[0]}(0))-\Lambda(\boldsymbol{\theta}^*)\right]\\&+\sum_{i'=0}^{i}\frac{L}{2}\left(1-\frac{\mu}{L}\right)^{i-i'}\|\Delta\mathbf{x}^{[i']}\|_2^2.\end{aligned} \tag{61}$$

Taking expectation on both sides and applying (56), (61) becomes

$$\mathbb{E}\left[\Lambda(\mathbf{x}_k^{[i+1]}(0))-\Lambda(\boldsymbol{\theta}^*)\right]\leq\left(1-\frac{\mu}{L}\right)^{i+1}\left[\Lambda(\mathbf{x}_k^{[0]}(0))-\Lambda(\boldsymbol{\theta}^*)\right]+\sum_{i'=0}^{i}A^{[i']}\max_k\mathbb{MSE}_k^{[i']}. \tag{62}$$

Finally, setting, $i=R-1$, taking the limit $R\to+\infty$ and using $(1-\mu/L)^R\to 0$, the proof is completed.

Appendix B
Proof of Theorem 2

Let $\mathbf{a}_k^{[t]}$ with $t=0,\cdots,RE-1$ be the parameter vector at user k such that $\mathbf{a}_k^{[t]}=\mathbf{x}_k^{[i]}(\tau)$ for $t=iE+\tau$. Define the following virtual sequences $\{\widetilde{\mathbf{a}}^{[t]}\}$ as

$$\widetilde{\mathbf{a}}^{[t]}=\sum_{k=1}^{K}\alpha_k\mathbf{a}_k^{[t]}+\sum_{k=1}^{K}\alpha_k\boldsymbol{\sigma}_k^{[t]}\mathbb{I}_{t\bmod E=0}, \tag{63}$$

where $\boldsymbol{\sigma}_k^{[t]}$ denotes the model distortion at the k-th user and the t-th iteration, which is upper bounded by $\mathbb{E}[\|\boldsymbol{\sigma}_k^{[t]}\|_2^2] \leq \max_{\forall i,k} \mathbb{MSE}_k^{[i]}$. Based on (63) and equations (2) and (10) in Section II of the revised manuscript, we have

$$\widetilde{\mathbf{a}}^{[t+1]} = \widetilde{\mathbf{a}}^{[t]} - \frac{\varepsilon}{\sum_{k=1}^{K} |\mathcal{D}_k|} \sum_{k=1}^{K} \sum_{\mathbf{d}_{k,l} \in \mathcal{D}_k} \nabla_{\mathbf{x}_k} \Theta(\mathbf{d}_{k,l}, \mathbf{x}_k)|_{\mathbf{x}_k = \mathbf{a}_k^{[t]}} + \sum_{k=1}^{K} \alpha_k \left(\boldsymbol{\sigma}_k^{[t+1]} \mathbb{I}_{t+1 \bmod E=0} - \boldsymbol{\sigma}_k^{[t]} \mathbb{I}_{t \bmod E=0} \right). \tag{64}$$

Since $\varepsilon = \frac{2}{\mu(\nu+iE+\tau)} \leq \frac{1}{4L}$ and according to [21, Lemma 1], we have

$$\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t+1]} - \boldsymbol{\theta}^*\|_2^2] \leq (1-\mu\varepsilon)\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \boldsymbol{\theta}^*\|_2^2] + \sum_{j=1}^{K} \alpha_j \mathbb{E}[\|\boldsymbol{\sigma}_j^{[t+1]} \mathbb{I}_{t+1 \bmod E=0} - \boldsymbol{\sigma}_j^{[t]} \mathbb{I}_{t \bmod E=0}\|_2^2] + \frac{2}{K} \sum_{k=1}^{K} \mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \mathbf{a}_k^{[t]}\|_2^2] + 6L\Gamma\varepsilon^2. \tag{65}$$

To further bound the right hand side of (65), we notice that for any (k,t),

$$\mathbb{E}[\|\boldsymbol{\sigma}_k^{[t+1]} \mathbb{I}_{t+1 \bmod E=0} - \boldsymbol{\sigma}_k^{[t]} \mathbb{I}_{t \bmod E=0}\|_2^2] \leq \mathbb{E}[\|\boldsymbol{\sigma}_k^{[t]}\|^2] \leq \max_{\forall i,k} \mathbb{MSE}_k^{[i]}. \tag{E6}$$

Moreover, based on [22, Lemma A.3], the term $\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \mathbf{a}_k^{[t]}\|_2^2]$ is upper bounded as

$$\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \mathbf{a}_k^{[t]}\|_2^2] \leq 4\varepsilon^2 G^2 E^2. \tag{66}$$

Consequently, the following recursive error bound is obtained

$$\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t+1]} - \boldsymbol{\theta}^*\|_2^2] \leq (1-\mu\varepsilon)\mathbb{E}[\|\widetilde{\mathbf{a}}_k^{[t]} - \boldsymbol{\theta}^*\|_2^2] + \varepsilon^2 C. \tag{67}$$

where C is defined in Theorem 2. It can be seen that the above inequality is the same as equation (B.1) in [22, Lemma A.3]. Following the induction procedure in Appendix B of [22], the non-recursive error bound is obtained

$$\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \boldsymbol{\theta}^*\|_2^2] \leq \frac{\max(4C/\mu^2, \nu\|\boldsymbol{\theta}^{[0]} - \boldsymbol{\theta}^*\|_2^2)}{t+\nu}. \tag{E9}$$

Multiplying $2L$ on both sides of and using the Lipschitz condition of Λ yield

$$\mathbb{E}\left[\Lambda(\widetilde{\mathbf{a}}^{[t]})\right] - \Lambda(\boldsymbol{\theta}^*) \leq 2L\mathbb{E}[\|\widetilde{\mathbf{a}}^{[t]} - \boldsymbol{\theta}^*\|_2^2] \leq \frac{2L\max(4C, \mu^2\nu\|\boldsymbol{\theta}^{[0]} - \boldsymbol{\theta}^*\|_2^2)}{\mu^2(t+\nu)}, \tag{68}$$

Substituting $\widetilde{\mathbf{a}}_k^{[t]} = \mathbf{x}_k^{[R]}(0)$ and $t = ER$ into (68), the proof is completed.

APPENDIX C
PROOF OF LEMMA 1

Define a surrogate function of $h(\mathbf{v})$ as

$$g(\mathbf{v},\mathbf{v}') = -\sum_{k=1}^{K}\frac{b_k|\mathbf{g}_k^H\mathbf{v}|^2}{\sigma_k^2} + \sum_{k=1}^{K}\frac{b_k\left(\mathbf{v}-\mathbf{v}'\right)^H\mathbf{g}_k\mathbf{g}_k^H\left(\mathbf{v}-\mathbf{v}'\right)}{\sigma_k^2}. \tag{69}$$

Since $g(\mathbf{v},\mathbf{v}') \geq h(\mathbf{v})$, $g(\mathbf{v}',\mathbf{v}') = h(\mathbf{v}')$, and $\nabla g(\mathbf{v}',\mathbf{v}') \geq \nabla h(\mathbf{v}')$, with any feasible $\mathbf{v}^{(0)}$, every limit point of the sequence $(\mathbf{v}^{(0)},\mathbf{v}^{(1)},\cdots)$ generated by the following iteration

$$\mathbf{v}^{(n+1)} = \arg\min_{\|\mathbf{v}\|_2^2\leq\beta}\ \max_{\mathbf{b}\in\Delta} g(\mathbf{v},\mathbf{v}^{(n)}) \tag{70}$$

is the KKT solution to (35). Therefore, to prove the Lemma, it remains to show that $U(\mathbf{v}')$ is the optimal solution to (70). Specifically, applying the quasi-concave-convex property of (70) and the general minimax theorem [36], we have

$$\min_{\|\mathbf{v}\|_2^2\leq\beta}\ \max_{\mathbf{b}\in\Delta} g(\mathbf{v},\mathbf{v}') = \max_{\mathbf{b}\in\Delta}\ \min_{\|\mathbf{v}\|_2^2\leq\beta} g(\mathbf{v},\mathbf{v}'). \tag{71}$$

Via the Lagrange multiplier method, it can be derived that

$$\arg\min_{||\mathbf{v}||_2^2\leq\beta} g(\mathbf{v},\mathbf{v}') = \frac{\sqrt{\beta}\mathbf{C}(\mathbf{v}')\mathbf{b}}{\left|\left|\mathbf{C}(\mathbf{v}')\mathbf{b}\right|\right|_2}. \tag{72}$$

Putting the above result into $g(\mathbf{v},\mathbf{v}')$, we have

$$g\left(\frac{\sqrt{\beta}\mathbf{C}(\mathbf{v}')\mathbf{b}}{\left|\left|\mathbf{C}(\mathbf{v}')\mathbf{b}\right|\right|_2},\mathbf{v}'\right) = -\Phi\left(\mathbf{v}',\mathbf{b}\right). \tag{73}$$

Therefore, the optimal solution of $\mathbf{v}$ to (71) (thus (70)) is $U(\mathbf{v}')$ and the proof is completed.

APPENDIX D
PROOF OF LEMMA 2

To prove this theorem, we first introduce the following lemma.

Lemma 4. *([40, Lemma 1.2.2]) If $h(\mathbf{x})$ is convex and twice differentiable, then $h(\mathbf{x})$ is Lipschitz smooth with constant L if and only if $\nabla^2 h(\mathbf{x}) \preceq L\,\mathbf{I}$.*

Based on Lemma 4 and since $\Xi(\mathbf{b})$ is convex and twice differentiable, it suffices to show $\nabla^2\Xi(\mathbf{b}) \preceq L_\Xi(\phi)\,\mathbf{1}_K$. In particular, according to (42), the Hessian matrix of $\Xi(\mathbf{b})$ is

$$\nabla^2\Xi(\mathbf{b}) = \frac{2\sqrt{\beta}\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right)}{\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2}} - \frac{2\sqrt{\beta}}{(\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2})^3}\times\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right)\left[\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right)\right]^H.$$

Due to $\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right)[\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right)]^H \succeq \mathbf{0}$, we can drop the last term to bound $\nabla^2\Xi(\mathbf{b})$ from above, which leads to

$$\nabla^2\Xi(\mathbf{b}) \preceq \frac{2\sqrt{\beta}}{\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2}} \times \mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\mathbf{b}\right) \preceq \frac{2\sqrt{\beta}\cdot\lambda_{\max}\left[\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right)\right]}{\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2}}\mathbf{I}_K, \tag{74}$$

where the second inequality follows from $\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right) \preceq \lambda_{\max}\left[\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right)\right]\mathbf{I}_K$.

Now, the only quantity in (74) that is dependent on $\mathbf{b}$ is $||\mathbf{C}\mathbf{b}||_2^2$. To get rid of such dependence, $||\mathbf{C}\mathbf{b}||_2^2$ is lower bounded by

$$||\mathbf{C}\mathbf{b}||_2^2 = \mathbf{b}^T\mathbf{C}^H\mathbf{C}\mathbf{b} \geq \lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right) \times ||\mathbf{b}||_2^2. \tag{75}$$

Finally, using $||\mathbf{b}||_2^2 \geq 1/K$ due to $\mathbf{b} \in \Delta$ and Cauchy-Schwarz inequality further leads to

$$||\mathbf{C}\mathbf{b}||_2^2 \geq \lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right)/K. \tag{76}$$

Replacing $||\mathbf{C}\mathbf{b}||_2^2$ in (74) with the right hand side of (76), we immediately obtain $\nabla^2\Xi(\mathbf{b}) \preceq L_\Xi(\phi)\,\mathbf{I}_K$.

Appendix E
Proof of Theorem 4

To prove part (i) of this theorem, notice that $\mathrm{Rank}\left(\mathbf{C}\right) = \mathrm{Rank}\left([\mathbf{g}_1,\cdots,\mathbf{g}_K]\right)$ due to the definition of $\mathbf{C}$ in (38). As $\mathrm{Rank}\left(\mathbf{C}^H\mathbf{C}\right) = \mathrm{Rank}\left(\mathbf{C}\right)$, $\mathrm{Rank}\left(\mathbf{C}^H\mathbf{C}\right) = K$ and $\lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right) > 0$. Putting $\lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right) > 0$ and $\phi = 0$ into (44), we obtain

$$L_\Xi(0) = \frac{2\sqrt{\beta}\left|\left|\mathrm{Re}\left(\mathbf{C}^H\mathbf{C}\right)\right|\right|_2}{\sqrt{\lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right)/K}} < +\infty. \tag{77}$$

Next, to prove part (ii), it can be seen that $\lambda_{\min}\left(\mathbf{C}^H\mathbf{C}\right) = 0$ if $\mathrm{Rank}\left([\mathbf{g}_1,\cdots,\mathbf{g}_K]\right) \neq K$. Putting this result and $\phi = 0$ into (44), we obtain $L_\Xi(0) = +\infty$. On the other hand, if $\phi > 0$, we must have $\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2} > 0$ due to $||\mathbf{C}\mathbf{b}||_2^2 \geq 0$. Putting this result into (44), we obtain $L_\Xi(\phi) < +\infty$ if $\phi > 0$.

Finally, to prove part (iii) of this theorem, we need the following lemma.

Lemma 5. $\Xi(\mathbf{b})$ *is bounded as* $\Phi(\mathbf{b}) \leq \Xi(\mathbf{b}) \leq \Phi(\mathbf{b}) + 2\sqrt{\beta}\,\phi$.

Proof. To prove the left inequality, notice that $||\mathbf{C}\mathbf{b}||_2 \leq \sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2}$. Putting this result into $\Xi(\mathbf{b})$ in (42), we immediately obtain $\Phi(\mathbf{b}) \leq \Xi(\mathbf{b})$. On the other hand, to prove the right inequality, we first compute

$$\Xi(\mathbf{b}) - \Phi(\mathbf{b}) = 2\sqrt{\beta}\Big(\sqrt{\phi^2+||\mathbf{C}\mathbf{b}||_2^2} - \sqrt{||\mathbf{C}\mathbf{b}||_2^2}\Big). \tag{78}$$

Then applying the identity $\sqrt{\phi^2+x}-\sqrt{x}=\frac{\phi^2}{\sqrt{\phi^2+x}+\sqrt{x}}\leq\phi$, where the inequality is due to the monotonic decreasing feature of $\phi^2/(\sqrt{\phi^2+x}+\sqrt{x})$ with respect to x, the right hand side of (78) is upper bounded as

$$2\sqrt{\beta}\Big(\sqrt{\phi^2+||\mathbf{Cb}||_2^2}-\sqrt{||\mathbf{Cb}||_2^2}\Big)\leq 2\sqrt{\beta}\,\phi. \tag{79}$$

Putting this result into (78), we have $\Xi(\mathbf{b})-\Phi(\mathbf{b})\leq 2\sqrt{\beta}\,\phi$. ■

Now, if an ϵ-optimal solution $\mathbf{b}'\in\Delta$ to $\mathcal{Q}_2$ is obtained with $\Xi(\mathbf{b}')-\Xi(\mathbf{b}^\diamond)\leq\epsilon$, then we must have

$$\Phi(\mathbf{b}')\leq\Xi(\mathbf{b}^\diamond)+\epsilon, \tag{80}$$

due to $\Phi(\mathbf{b})\leq\Xi(\mathbf{b})$ from the first inequality of Lemma 5. On the other hand, taking the minimum on both sides of the second inequality of Lemma 5, we have

$$\min_{\mathbf{b}\in\Delta}\ \Xi(\mathbf{b})\leq\min_{\mathbf{b}\in\Delta}\ \Phi(\mathbf{b})+2\sqrt{\beta}\,\phi. \tag{81}$$

Putting $\min_{\mathbf{b}\in\Delta}\ \Xi(\mathbf{b})=\Xi(\mathbf{b}^\diamond)$ and $\min_{\mathbf{b}\in\Delta}\ \Phi(\mathbf{b})=\Phi(\mathbf{b}^*)$ into (81), (81) becomes $\Xi(\mathbf{b}^\diamond)\leq\Phi(\mathbf{b}^*)+2\sqrt{\beta}\,\phi$. Combining this result with (80) leads to $\Phi(\mathbf{b}')\leq\Phi(\mathbf{b}^*)+2\sqrt{\beta}\,\phi+\epsilon$, and part (iii) of this theorem is proved.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nat.*, vol. 521, pp. 436–444, May 2015.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May–Jun. 2020.

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

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

[5] S. Huang, S. Wang, R. Wang, M. Wen, and K. Huang, "Reconfigurable intelligent surface assisted mobile edge computing with heterogeneous learning tasks," *IEEE Trans. Cognitive Commun. Networking*, vol. 7, no. 2, pp. 369–382, Jun. 2021.

[6] Google Research, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[7] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge AI: Algorithms and systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2167–2191, Fourthquarter 2020.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, Fort Lauderdale, Florida, Apr. 2017.

[9] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

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

[11] T. Zen, O. Semiari, M. Mozaffari, M. Chen, W. Saad, and M. Bennis, "Federated learning in the sky: Joint power allocation and scheduling with UAV swarms," in *Proc. IEEE ICC*, Dublin, Ireland, Jun. 2020.

[12] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, Mar. 2020.

[13] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[14] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, Mar. 2020.

[15] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2020.

[16] 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.

[17] H. Guo, A. Liu, and V. K. N. Lau, "Analog gradient aggregation for federated learning over wireless networks: Customized design and convergence analysis," *IEEE IoT Journal*, vol. 8, no. 1, pp. 197–210, Jan. 2021.

[18] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Toward optimal rate-delay tradeoff for computation over multiple access channel," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4335–4346, Jul. 2021.

[19] L. Chen, N. Zhao, Y. Chen, X. Qin, and F. R. Yu, "Computation over MAC: Achievable function rate maximization in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5446–5459, Sep. 2020.

[20] H. Hellström, J. Mairton, B. da Silva Jr, V. Fodor, and C. Fischione, "Wireless for machine learning," [Online]. Available: https://arxiv.org/abs/2008.13492, 2020.

[21] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proc. ICLR*, Addis Ababa, Ethiopia, Apr. 2020.

[22] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, Jun. 2021.

[23] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Apr. 2016.

[24] V. Venkateswaran and A. J. van der Veen, "Analog beamforming in MIMO communications with phase shift networks and online channel estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4131–4143, Aug. 2010.

[25] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, pp. 3337, Oct. 2018.

[26] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. CoRL*, 2017, pp. 1–16.

[28] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. 1380–1405, May 2012.

[29] A. Beck, "On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes," *SIAM J. Optimiz.*, vol. 25, no. 1, pp. 185–209, 2015.

[30] X. Zhang, *Matrix Analysis and Applications*. Beijing, China: Tsinghua Univ. Press, 2004.

[31] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imaging Sci.*, vol. 1, no. 3, pp. 248–272, 2008.

[32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[33] S. Wang, L. Cheng, M. Xia, and Y.-C. Wu, "Massive MIMO multicast beamforming via accelerated random coordinate descent," in *Proc. IEEE ICASSP'19*, Brighton, UK, May 2019, pp. 4494–4498.

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[35] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[36] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–176, 1958.

[37] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.,* vol. 103, no. 1, pp. 127–152, May 2005.

[38] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights", *J. Mach. Learn. Res.*, vol. 17, no. 153, pp. 1–43, Sep. 2016.

[39] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[40] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course.* Applied Optimization. Springer, 2004.

[41] L. Condat, "Fast projection onto the simplex and the l_1 ball," *Math. Program.*, vol. 158, no. 1, pp. 575–585, Jul. 2016.

[42] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[43] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *Proc. IEEE ICRA*, Xi'an, China, May–Jun. 2021, pp. 953 959.

[44] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun , "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[45] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 683 –711, Feb. 2021.

[46] S. Wang, D. Li, R. Wang, Q. Hao, Y.-C. Wu, and D. W. K. Ng, "Unit-modulus wireless federated learning via penalty alternating minimization," in *Proc. IEEE GLOBECOM*, Madrid, Spain, Dec. 2021.