A Graph Neural Network Learning Approach to Optimize RIS-Assisted Federated Learning

Zixin Wang, *Student Member*, *IEEE*, Yong Zhou, *Senior Member*, *IEEE*, Yinan Zou, *Student Member*, *IEEE*, Qiaochu An, *Student Member*, *IEEE*, Yuanming Shi, *Senior Member*, *IEEE*, and Mehdi Bennis, *Fellow*, *IEEE*

***Abstract*—Over-the-air federated learning (FL) is a promising privacy-preserving edge artificial intelligence paradigm, where over-the-air computation enables spectral-efficient model aggregation by achieving simultaneous communication and aggregation. However, due to limited transmit power, the performance of over-the-air FL is limited by the device with the worst channel condition toward the edge server. In this paper, we leverage reconfigurable intelligent surface (RIS) to mitigate the communication bottleneck of over-the-air FL and explicitly characterize the corresponding convergence upper bound. The convergence analysis illustrates the detrimental impact of the accumulated aggregation error over all rounds and inspires us to formulate a time-average transmission distortion minimization problem by jointly optimizing the transceiver and RIS phase-shifts. To reduce the computation complexity and enhance the model aggregation accuracy, we develop a graph neural network (GNN) based learning algorithm to directly map channel coefficients to the optimized network parameters. By exploiting permutation equivalence and invariance properties of graphs, the parameter dimension of the proposed algorithm is independent of the number of edge devices, which reduces the computational complexity and improves the algorithmic scalability. Simulations show that the proposed algorithm speeds up the computation by three orders of magnitude compared to the baselines, while achieving performance superiority and algorithmic robustness.**

***Index Terms*—Federated learning, graph neural network, reconfigurable intelligent surface, over-the-air computation.**

I. Introduction

The advancement of artificial intelligence (AI) is boosting the development of many intelligent applications (e.g., virtual reality, smart industry, autonomous driving) in future wireless networks [1]. Numerous raw data generated by edge devices can be exploited for intelligence distillation at the network edge [2]. However, as the data privacy concern increases, it is undesirable to transfer raw data from edge devices to an edge server. To alleviate this privacy concern, federated learning (FL), as a privacy-preserving edge AI paradigm, has recently been proposed [3]. In particular, FL allows multiple edge devices to cooperatively train a global model under the coordination of an edge server without disclosing any raw data of edge devices.

To implement FL over wireless networks, high-dimensional model transmission between devices and server over fading channels is required. With limited radio resource, designing communication-efficient model/gradient transmission schemes has attracted much attention [4]–[12]. Specifically, the authors in [4] proposed to quantize the local models as binary sequences to alleviate the communication load. The authors in [5] exploited the model sparsification technique to reduce the bandwidth requirement. Device selection is another useful technique that enables FL over bandwidth-limited wireless networks. The authors in [6] developed a multi-armed bandit device selection strategy to minimize the training latency of FL. Transmission reliability was further considered for device selection in [7] to enhance the learning performance. The authors in [8] proposed robust FL design for both expectation-based and worst-case noisy models. Momentum FL was proposed in [9] to accelerate the convergence and in turn reduce the communication overhead. Moreover, joint optimization of device selection and sub-channel allocation was studied in [10] to achieve adaptive model transmission. The authors in [11] proposed to achieve energy-efficient FL by jointly optimizing communication and computation resources. All the above works utilized orthogonal multiple access schemes to achieve reliable model aggregation. A limited number of resource blocks restricts the amount of edge devices participating in FL training and is the performance bottleneck of FL over wireless networks.

To alleviate this issue, over-the-air computation (AirComp) is an efficient technique for enabling spectrum-efficient uplink model aggregation in FL [13]–[17]. With AirComp, multiple edge devices that share the same radio channel simultaneously transmit their models/gradients. Because of the inherent signal superposition property, the edge server directly receives an aggregation of the concurrently transmitted models, which can be adopted to update the global model. To mitigate transmission distortion due to channel fading and receiver noise, various studies were proposed from the perspectives of transmit power control [13], [14], receive beamforming design [15], device selection [16], and bandwidth allocation [17]. AirComp-based FL requires the signals transmitted by edge devices to be aligned at the edge server. Because of limited transmit power,

Manuscript received May 17, 2022; revised Nov. 27, 2022; accepted Jan. 16, 2023. The work of Yong Zhou was supported by the National Natural Science Foundation of China (NSFC) under Grants U20A20159, 62001294, and 61971286. The work of Yuanming Shi was supported in part by the Natural Science Foundation of Shanghai under Grant No. 21ZR1442700 and Shanghai Rising-Star Program under Grant No. 22QA1406100. (Corresponding authors: Yong Zhou and Yuanming Shi.)

Z. Wang is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China, also with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (email: wangzx2@shanghaitech.edu.cn).

Y. Zhou, Y. Zou, Q. An, and Y. Shi are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (email: {zhouyong, zouyn, anqch, shiym}@shanghaitech.edu.cn).

M. Bennis is with the Centre for Wireless Communication, University of Oulu, Finland (email: mehdi.bennis@oulu.fi).

the edge device with the worst channel condition toward the edge server determines the signal alignment error, which in turn affects the performance of AirComp-based FL.

Reconfigurable intelligent surface (RIS) provides an effective solution to alleviate the performance bottleneck of AirComp-based FL by adaptively reconfiguring the propagation environment [18]–[26]. RIS generally consists of many passive reflection elements, which can be dynamically adjusted to reflect the incident signal in the desired manner, thereby facilitating signal alignment at the receiver. RIS has been leveraged to enhance the differential privacy [18], performance robustness [19], [20], energy efficiency [21], [22], and spectrum efficiency [23]. In [24], the authors proposed an RIS-assisted AirComp-based FL framework and developed an effective resource allocation algorithm based on the theoretical convergence analysis. The authors in [25] tackled the straggler issue in AirComp-based FL by jointly optimizing device selection, AirComp transceiver, and RIS phase-shifts. However, all these works studied the resource allocation for each communication round but ignored a key feature of FL, i.e., FL involves multiple communication rounds and its performance is determined by the model aggregation errors accumulated over all rounds. Without optimizing resource allocation from a long-term perspective may degrade the learning performance. Moreover, although deploying RIS can enhance the learning performance, it incurs a high computation complexity for joint AirComp transceiver and RIS phase-shift optimization. These two issues motivate this paper.

Deep learning has recently been adopted to achieve computation-efficient resource allocation by learning a mapping from channel state information (CSI) to system design [27]–[32]. In particular, graph neural networks (GNN) has a great potential in improving the algorithmic scalability for resource allocation [33]–[35]. By leveraging the permutation equivalence and invariance, the authors in [33] developed a random edge GNN for power allocation for wireless networks with frequency reuse, which was extended to the scenario with multiple antennas in [34]. The authors in [35] developed a GNN-based low-complexity resource allocation framework for joint link scheduling, channel allocation, and power control. Although GNN has been applied to maximize the network utility, the salient feature of GNN (i.e., permutation equivalence and invariance) has not been exploited to support efficient resource allocation for over-the-air FL.

In this paper, we study an RIS-assisted over-the-air FL, where AirComp and RIS are adopted to achieve fast and accurate uplink model aggregation, respectively. We aim to develop a scalable and computation-efficient algorithm that jointly optimizes the AirComp transceiver and RIS phase-shifts to enhance the performance of FL. Accomplishing such a goal faces the following challenges. First, the metric for characterizing the performance of FL in terms of communication parameters is not directly available and can only be obtained by conducting rigorous convergence analysis for the specifically designed wireless FL system. Second, the performance of FL is determined by the communication errors accumulated over all communication rounds, and hence should be evaluated from a long-term perspective. This challenging issue is further complicated by the average transmit power constraint. Third, to enhance the FL performance, the AirComp transceiver and RIS phase-shifts should be jointly optimized. Solving such a joint optimization problem typically requires an alternating optimization algorithm, which is computationally expensive. To address these challenges, we conduct a rigorous convergence analysis and formulate a joint optimization problem, followed by developing a novel GNN-based learning algorithm to achieve the efficient AirComp transceiver and RIS phase-shifts design. We summarize the main contributions of this paper as follows.

- We derive the convergence upper bound for RIS-assisted over-the-air FL as a function of the average norm of the global gradient, where the aggregation errors accumulated over all communication rounds are taken into account. The convergence analysis motivates us to formulate a time-average transmission distortion minimization problem, which requires AirComp transceiver and RIS phase-shifts to be jointly optimized. Such a non-convex joint optimization problem is solved by developing an alternating optimization algorithm, which exploits the time-sharing feature of the formulated problem.
- To reduce the computation complexity and enhance the model aggregation accuracy, we develop a GNN-based learning algorithm that directly maps the channel coefficients to the optimized design of AirComp transceiver and RIS phase-shifts. The proposed GNN framework captures the intricate interaction among multiple edge devices, RIS, and edge server, which enables the joint optimization for signal alignment and noise suppression. The permutation equivalence and permutation invariance properties of GNN enhance the scalability of the developed learning framework. Moreover, the parameter dimension of the proposed GNN-based learning algorithm is independent of the number of edge devices, which reduces the computational complexity and further improves the algorithmic scalability.
- Simulations show the excellence of the proposed GNN-based learning algorithm from the perspectives of test accuracy, robustness versus different system parameters, as well as computational efficiency. By exploiting the unique features of GNN to jointly optimize the AirComp transceiver and RIS phase-shifts, the proposed GNN-based learning algorithm achieves a close performance to the error-free transmission and significantly outperforms the optimization-based algorithm in terms of both learning performance and computation complexity.

The remaining of this paper is organized as follows. Section II presents the system model. The convergence analysis and problem formulation are presented in Section III. We propose an alternating optimization algorithm in Section IV. Section V presents a GNN-based learning framework. Simulation results are given in Section VI. Finally, Section VII concludes this paper.

Notations: We use italic, boldface lower-case, and boldface upper-case letters to represent scalar, vector, and matrix, respectively. Mathematical operators $\dagger$, $\mathrm{diag}(\cdot)$, $(\cdot)^{\mathrm{T}}$, $(\cdot)^{\mathrm{H}}$, $\mathrm{Tr}(\cdot)$,

TABLE I
DEFINITION OF MAIN NOTATIONS

Notation	Definition
Ω	Dimension of model parameters
ξ	Upper bound of local mini-batch gradient
M	Total number of training samples
$\boldsymbol{\Upsilon}_k(t)$	Local gradient of device k at the t-th round
$F_k(\boldsymbol{w})$	Local loss function at device k
K	Number of edge devices
γ	Learning rate
$\eta(t)$	Denoising factor at the $t-$th round
$\theta_n(t)$	Phase-shift of the n-th element at the $t-$th round
$p_i(t)$	Transmit power of device i at the $t-$th round
Γ	Upper bounds the variance of Ω elements of $\boldsymbol{\Upsilon}_k$, $\forall\, k \in \mathcal{K}$
$h_i^{\mathrm{c}}(t)$	Combined channel response between device i and the edge server
T	Total number of communication rounds
$\boldsymbol{z}_k^d$	Representation vector of node k at layer d

$\mathbb{E}(\cdot)$, $|\cdot|$, and $\|\cdot\|$ denote the conjugate operation, diagonal matrix, transpose, Hermitian transpose, trace, statistical expectation, the cardinality of a set or the absolute value operation, and the Euclidean norm, respectively. $\mathbb{R}$ and $\mathbb{C}$ denote the real and complex spaces, respectively. The frequently used notations are listed in Table I.

II. System Model

A. RIS-Assisted Over-the-Air FL

As shown in Fig. 1, we consider an RIS-assisted over-the-air FL network, where a single-antenna edge server coordinates a set $\mathcal{K} = \{1, \ldots, K\}$ of K single-antenna devices to train a global model with the assistance of an RIS. Each edge device $k \in \mathcal{K}$ owns a local dataset denoted by $\mathcal{D}_k = \{(\boldsymbol{x}_{km}, \boldsymbol{y}_{km}) \mid 1 \le m \le M_k\}$ for local model training, where $(\boldsymbol{x}_{km}, \boldsymbol{y}_{km})$ denotes the m-th input feature and label pair at device k, and M_k denotes the number of training samples available at device k. We assume that the training datasets at different edge devices are independent and identically distributed (i.i.d.), and have the same number of training samples, i.e., $M_k = M_j, \forall\ k, j \in \mathcal{K}$, as in [24], [36]. We aim to find the optimal model parameter vector $\boldsymbol{w}^* \in \mathbb{R}^{\Omega}$ that minimizes the global loss function $F(\boldsymbol{w})$, i.e., $\min_{\boldsymbol{w}} F(\boldsymbol{w}) = \frac{1}{M}\sum_{k\in\mathcal{K}} M_k F_k(\boldsymbol{w}) = \frac{1}{K}\sum_{k\in\mathcal{K}} F_k(\boldsymbol{w})$, where $M = \sum_{k\in\mathcal{K}} M_k$ is the total number of training samples and $F_k(\boldsymbol{w})$ is local loss function at device k. In each round $t = 1, \ldots, T$, the following three steps are performed.

- *Global model dissemination*: Each edge device receives global model $\boldsymbol{w}(t-1)$ through the downlink channel from the edge server at the beginning of round t. Since the edge server transmits with a much greater power than edge devices, it is reasonable to assume that the distortion of the global model at each device is negligible, as in [24], [37].
- *Local model update*: According to the received global model $\boldsymbol{w}(t-1)$, each edge device $k \in \mathcal{K}$ evaluates its local stochastic gradient $\boldsymbol{\Upsilon}_k(t) = \nabla F_k\left(\boldsymbol{w}(t-1); \mathcal{B}_k^{FL}\right)$, where $\mathcal{B}_k^{FL} \subset \mathcal{D}_k$ denotes the mini-batch containing $|\mathcal{B}_k^{FL}|$ randomly sampled data samples.
- *Local model aggregation*: As the edge server only requires the arithmetic mean of local gradients (i.e., $\boldsymbol{\Upsilon}(t) = \frac{1}{K}\sum_{k\in\mathcal{K}} \boldsymbol{\Upsilon}_k(t)$) for the global model update (i.e., $\boldsymbol{w}(t)$), we adopt AirComp to achieve low-latency uplink gradient aggregation. With AirComp, the concurrently transmitted local gradients $\{\boldsymbol{\Upsilon}_k(t)\}_{k\in\mathcal{K}}$ from K edge devices can be added over-the-air, and the edge server can directly receive a summation of these gradients. Because of random channel fading and receiver noise, the edge server recovers a noisy estimation of the average gradient $\boldsymbol{\Upsilon}(t)$. The achievable accuracy of local model aggregation via AirComp is limited by the worst channel among all device-server links [24]. To tackle this issue, we leverage an RIS with N reflection elements to mitigate the communication bottleneck and in turn improve the gradient aggregation accuracy. By denoting the recovered noisy estimation of $\boldsymbol{\Upsilon}(t)$ as $\hat{\boldsymbol{\Upsilon}}(t)$, the edge server updates the global model as

$$\boldsymbol{w}(t) = \boldsymbol{w}(t-1) - \gamma \hat{\boldsymbol{\Upsilon}}(t), \tag{1}$$

where γ denotes the learning rate.

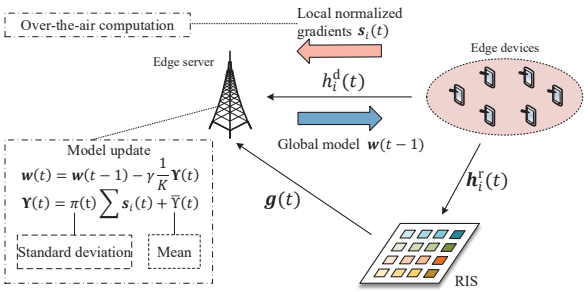


Fig. 1. Illustration of model aggregation in an RIS-assisted over-the-air FL system.

B. Signal Model

To optimize RIS-assisted over-the-air FL, we describe the signal model of the uplink gradient aggregation. After the local training, each edge device $k \in \mathcal{K}$ calculates the mean $\bar{\Upsilon}_k(t)$ and variance $\pi_k^2(t)$ of local gradient $\boldsymbol{\Upsilon}_k(t)$ as $\bar{\Upsilon}_k(t) = \frac{1}{\Omega}\sum_{j=1}^{\Omega} \Upsilon_{k,j}(t)$ and $\pi_k^2(t) = \frac{1}{\Omega}\sum_{j=1}^{\Omega}\left(\Upsilon_{k,j}(t) - \bar{\Upsilon}_k(t)\right)^2$, respectively, where $\Upsilon_{k,j}(t)$ is the j-th element of $\boldsymbol{\Upsilon}_k(t)$. After receiving the local statistics $\{\bar{\Upsilon}_k(t), \pi_k^2(t)\}$, the edge server computes the global mean and variance as $\bar{\Upsilon}(t) = \frac{1}{K}\sum_{j=1}^{K} \bar{\Upsilon}_k(t)$ and $\pi^2(t) = \frac{1}{K}\sum_{j=1}^{K} \pi_k^2(t)$, respectively, and then broadcasts them to each edge device k for normalization of $\boldsymbol{\Upsilon}_k(t)$ as in [25]. The signal to be transmitted by edge device k is normalized as $\boldsymbol{s}_k(t) = \frac{\boldsymbol{\Upsilon}_k(t) - \bar{\Upsilon}_k(t)}{\pi(t)}$, where $\mathbb{E}[\boldsymbol{s}_k(t)] = 0$ and $\mathbb{E}[\boldsymbol{s}_k(t)\boldsymbol{s}_k(t)^{\mathrm{T}}] = \boldsymbol{I}_{\Omega}$. We assume that the gradient statistics $\{\bar{\Upsilon}_k(t), \pi_k^2(t)\}$ are transmitted in an error-free manner as the size of the gradient statistics is generally much smaller than that of the gradient vector [25].

Let $\boldsymbol{g}(t) \in \mathbb{C}^N$, $\boldsymbol{h}_i^{\mathrm{r}}(t) \in \mathbb{C}^N$, and $h_i^{\mathrm{d}}(t) \in \mathbb{C}$ be the channel response vector between the RIS and edge server, channel response vector between device i and RIS, and channel response between device i and edge server, respectively. All

channel responses are assumed to obey block-fading [38]. With synchronized transmissions, the signal received at the edge server in round t is

$$\boldsymbol{y}(t)=\sum\nolimits_{i=1}^{K}\left[h_i^{\mathrm{d}}(t)+\boldsymbol{g}(t)\mathrm{diag}\left(\boldsymbol{v}(t)\right)\boldsymbol{h}_i^{\mathrm{r}}(t)\right]b_i(t)\boldsymbol{s}_i(t)+\boldsymbol{n}(t), \tag{2}$$

where $b_i(t)$ denotes the transmit scalar of device i, $\boldsymbol{v}(t)=[\beta e^{j\theta_1(t)},\beta e^{j\theta_2(t)},\ldots,\beta e^{j\theta_N(t)}]^{\mathrm{T}}$ denotes the phase-shift vector of RIS with β being the reflection amplitude and $\theta_n(t)$ being the reflection phase of the n-th element, and $\boldsymbol{n}(t)\sim\mathcal{CN}(0,\sigma^2\boldsymbol{I})$ denotes the *additive white Gaussian noise* (AWGN) with variance $\sigma^2\boldsymbol{I}$. For simplicity, we set β to 1, as in [24]–[26].

For notational ease, let $h_i^{\mathrm{c}}(t)=h_i^{\mathrm{d}}(t)+\boldsymbol{g}(t)\mathrm{diag}(\boldsymbol{v}(t))\boldsymbol{h}_i^{\mathrm{r}}(t)$ be the combined channel response between device i and edge server. The transmit power of device i is denoted as $p_i(t)\geq 0$. As many existing channel estimation strategies [39]–[42] can be adopted to effectively estimate the CSI for RIS-assisted wireless networks, we assume that the perfect CSI is available in this paper. The communication overhead due to channel estimation can be ignored as the pilot length is generally much smaller than the size of the gradient vector, as in [24]. By setting $b_i(t)=\frac{\sqrt{p_i(t)}(h_i^{\mathrm{c}}(t))^{\dagger}}{|h_i^{\mathrm{c}}(t)|}$, the received signal in (2) is $\boldsymbol{y}(t)=\sum_{i=1}^{K}\left(|h_i^{\mathrm{c}}(t)|\sqrt{p_i(t)}\boldsymbol{s}_i(t)\right)+\boldsymbol{n}(t)$. Besides, with an average power budget $\bar{P}_k,\forall\, k\in\mathcal{K}$, we have $\frac{1}{T}\sum_{t=1}^{T}p_k(t)\leq\bar{P}_k,\forall\, k\in\mathcal{K}$. Upon receving signal $\boldsymbol{y}(t)$, the edge server recovers the arithmetic mean of $\boldsymbol{s}_i(t)$ via applying the denoising factor $\eta(t)>0$ as follows

$$\hat{\boldsymbol{s}}(t)=\frac{\boldsymbol{y}(t)}{\sqrt{\eta(t)}}=\sum_{i\in\mathcal{K}}\frac{\sqrt{p_i(t)}|h_i^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}\boldsymbol{s}_i(t)+\frac{\boldsymbol{n}(t)}{\sqrt{\eta(t)}}. \tag{3}$$

As $\boldsymbol{\Upsilon}_k(t)=\pi(t)\boldsymbol{s}_k(t)+\bar{\Upsilon}(t)$ and $\boldsymbol{s}(t)=\sum_{k\in\mathcal{K}}\boldsymbol{s}_k(t)$, the received stochastic gradient is $\hat{\boldsymbol{\Upsilon}}(t)=\frac{1}{K}\pi(t)\left(\hat{\boldsymbol{s}}(t)-\boldsymbol{s}(t)\right)+\boldsymbol{\Upsilon}(t)$ and the induced error in the uplink is

$$\bar{\boldsymbol{e}}(t)=\frac{1}{K}\left(\hat{\boldsymbol{s}}(t)-\boldsymbol{s}(t)\right). \tag{4}$$

By substituting (3) into (4), we have

$$\bar{\boldsymbol{e}}(t)=\frac{1}{K}\sum_{i\in\mathcal{K}}\left(\frac{\sqrt{p_i(t)}|h_i^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}-1\right)\boldsymbol{s}_i(t)+\frac{1}{K}\frac{\boldsymbol{n}(t)}{\sqrt{\eta(t)}}. \tag{5}$$

As can be observed, the aggregation accuracy of the local gradients depends on the transmit power, the channel coefficients, the RIS phase-shifts, the denoising factor, and the receiver noise. We characterize the impact of these parameters on FL in the following.

III. Convergence Analysis and Problem Formulation

In this section, we derive the upper bound of the time-average norm of the global gradient for RIS-assisted over-the-air FL and formulate an upper bound minimization problem.

A. Basic Assumptions

We first state several assumptions to facilitate the analysis.

Assumption 1. *There always exists some constant $F(\boldsymbol{w}^*)$ that lower bounds the global loss function, i.e., $F(\boldsymbol{w})\geq F(\boldsymbol{w}^*)$, $\forall\, \boldsymbol{w}$.*

Assumption 2. *$F_k(\boldsymbol{w})$ is continuously differentiable and is smooth with non-negative constant $S\geq 0$. The gradient $\nabla F_k(\boldsymbol{w})$ is also Lipschitz continuous with constant S, i.e.,*

$$\|\nabla F_k(\boldsymbol{w})-\nabla F_k(\boldsymbol{w}')\|_2\leq S\|\boldsymbol{w}-\boldsymbol{w}'\|_2,\ \forall\, \boldsymbol{w},\boldsymbol{w}'. \tag{6}$$

Assumption 3. *The local mini-batch gradient $\boldsymbol{\Upsilon}_k$ is an unbiased estimate of $\nabla F_k(\boldsymbol{w})$ with a bounded variance, bounded with a constant $\xi\geq 0$, i.e., $\mathbb{E}\left[\boldsymbol{\Upsilon}_k\right]=\nabla F_k(\boldsymbol{w})$ and $Var(\boldsymbol{\Upsilon}_k)=\mathbb{E}[\|\boldsymbol{\Upsilon}_k-\nabla F_k(\boldsymbol{w})\|_2^2]\leq\xi^2$, where $\xi\geq 0$ is a constant.*

Assumption 4. *There is a constant $\Gamma\geq 0$ that upper bounds the variance of Ω elements of $\boldsymbol{\Upsilon}_k$.*

These assumptions are standard in the existing studies of stochastic optimization [24], [25], [43]. With the above assumptions, we obtain Lemma 1.

Lemma 1. *The expected norm of the induced error is upper bounded as*

$$\mathbb{E}[\|\boldsymbol{e}(t)\|_2^2]\leq\Omega\frac{\Gamma(K+1)}{K^2}\mathrm{Er}(t), \tag{7}$$

where

$$\mathrm{Er}(t)=\sum_{k=1}^{K}\left(\frac{\sqrt{p_k(t)}|h_i^{\mathrm{d}}(t)+\boldsymbol{g}(t)\mathrm{diag}(\boldsymbol{v}(t))\boldsymbol{h}_i^{\mathrm{r}}(t)|}{\sqrt{\eta(t)}}-1\right)^2+\frac{\sigma^2}{\eta(t)}.$$

Proof. See Appendix A. □

According to Lemma 1, we derive the upper bound of $\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{w}(t))\|_2^2$ in Theorem 1.

Theorem 1 (Convergence). *With Assumptions 1-4, if $\gamma<\frac{1}{2S}$ and after T rounds, we bound the time-average norm of the global gradient as*

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{w}(t))\|_2^2\leq\underbrace{\frac{2(F(\boldsymbol{w}(0)-F(\boldsymbol{w}^*)))}{\gamma(1-2\gamma S)T}}_{\text{Initial optimality gap}}+\underbrace{\frac{2S\gamma\xi^2}{K(1-2\gamma S)}}_{\text{Gradient variance induced gap}}+\underbrace{\frac{1+2\gamma S}{1-2\gamma S}\frac{1}{T}\sum_{t=0}^{T-1}\underbrace{\mathbb{E}\left[\|\bar{\boldsymbol{e}}(t)\|_2^2\right]}_{\text{Instantaneous error}}}_{\text{Time-average error}}.$$

Proof. See Appendix B. □

According to Lemma 1 and Theorem 1, we have the following observations.

- **Upper bound decomposition.** The upper bound in Theorem 1 consists of the initial optimality gap, gradient variance induced gap, and time-average error. The initial optimality gap approaches zero as $T\to\infty$. The gradient variance induced gap is determined by the learning rate, Lipschitz constant, number of edge devices, and the variance of the local gradient. Specifically, enlarging

the size of the mini-batch reduces the gradient variance induced gap because the local mini-batch gradient can be bounded with a smaller ξ when the batch size is larger. Besides, having a greater number of participating edge devices reduces the gap as well. Once these parameters are fixed, the gradient variance induced gap is a constant. Thus, the time-average error dominates the upper bound when parameters γ, ξ, and S are fixed and T is large. This observation motivates us to minimize the time-average error, which further enhances the FL convergence performance[1].

- **Necessity of jointly optimizing AirComp transceiver and RIS phase-shifts.** According to Lemma 1, the time-average error depends on the qualities of the combined channels, which can be enhanced by optimizing the RIS phase-shifts. In addition, the transmit power of edge devices and the denoising factor determine the trade-off between signal alignment and noise reduction. Hence, to minimize the time-average error and in turn enhance the learning performance, the RIS phase-shifts, the transmit power, and the denoising factor should be jointly optimized.

Observing the detrimental effect of transmission errors on the convergence of FL, a joint optimization problem shall be formulated to minimize the time-average error.

B. Problem Formulation

We aim to propose a scalable and efficient algorithm that minimizes the time-average error. The corresponding optimization problem is

$$\begin{aligned}\mathscr{P}_0: \min_{\substack{\{\boldsymbol{p}(t)\}\\ \{\boldsymbol{v}(t),\eta(t)\}}} \ & \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{k\in\mathcal{K}}\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}-1\right)^2+\frac{\sigma^2}{\eta(t)}\right]\\ \text{s.t.} \quad & \frac{1}{T}\sum_{t=1}^{T}p_k(t)\le \bar{P}_k, \forall\, k\in\mathcal{K},\\ & |\boldsymbol{v}_j(t)|=1, \forall\, j=1,\ldots,N, \ \ \forall\, t=1,\ldots,T,\end{aligned} \tag{8}$$

where $\boldsymbol{p}(t)=[p_1(t),\ldots,p_K(t)]$ the transmit power vector of edge devices. It is challenging to jointly optimize these variables for problem $\mathscr{P}_0$ because of the following reasons. First, a large number of phase shifts leads to a high-dimensional non-convex optimization problem. Second, the transmit power $\{p_k(t)\}_{k\in\mathcal{K}}$, the denoising factor $\eta(t)$, and the phase-shift vector $\{\theta_j(t)\}_{j=1}^N$ are coupled in the non-convex objective function of problem $\mathscr{P}_0$. Third, the unit modulus constraints of problem $\mathscr{P}_0$ are non-convex, which further increases the difficulty of optimization. To solve problem $\mathscr{P}_0$, we first develop an alternating optimization algorithm in Section IV, and then propose a novel GNN-based learning algorithm in Section V.

[1]The convergence result in Theorem 1 can be extended to the scenario with non-i.i.d. data by defining a metric to characterize the divergence between the global gradient and local gradient, and deriving an upper bound for the accumulated difference between the global model and the individual local model in terms of the norm of the global gradient and the metric that characterizes the gradient divergence, as in [44]

IV. Alternating Optimization Algorithm

A. AirComp Transceiver Optimization

As problem $\mathscr{P}_0$ satisfies the time-sharing condition [45], strong duality holds and thus the Lagrangian-duality method can be applied. The Lagrangian function of problem $\mathscr{P}_0$ is

$$\begin{aligned}&\mathcal{L}(\{\boldsymbol{p}(t),\boldsymbol{v}(t),\eta(t),\lambda_k\})\\ =&\frac{1}{T}\sum_{t=1}^{T}\mathrm{Er}(t)+\sum_{k\in\mathcal{K}}\lambda_k\left(\frac{1}{T}\sum_{t=1}^{T}p_k(t)-\bar{P}_k\right),\end{aligned}$$

where $\lambda_k\ge 0$ denotes the dual variable with respect to the transmit power constraint of device k. Hence, the Lagrangian dual function is given by

$$G(\{\lambda_k\})=\min_{\{\boldsymbol{p}(t),\boldsymbol{v}(t),\eta(t)\}}\mathcal{L}(\{\boldsymbol{p}(t),\boldsymbol{v}(t),\eta(t),\lambda_k\}), \tag{9}$$

and the Lagrangian dual problem is given by

$$\max_{\{\lambda_k\ge 0\}} \ G(\{\lambda_k\}),\ \forall\, k\in\mathcal{K}. \tag{10}$$

As the strong duality holds, problem $\mathscr{P}_0$ can be equivalently solved by maximizing $G(\{\lambda_k\})$. Note that $G(\{\lambda_k\})$ is obtained by solving the minimization problem in (9) for given $\{\lambda_k \mid k\in\mathcal{K}\}$. We decompose problem (9) into multiple subproblems as follows

$$\min_{\substack{\{\boldsymbol{p}(t),\boldsymbol{v}(t),\}\\ \eta(t),\lambda_k}}\sum_{k\in\mathcal{K}}\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}-1\right)^2+\frac{\sigma^2}{\eta(t)}+\sum_{k\in\mathcal{K}}\lambda_k p_k(t). \tag{11}$$

When the phase-shift vector $\boldsymbol{v}(t)$ is fixed, the transmit power and the denoising factor can be designed as follows [38]

$$p_k^*(t)=\left(\frac{\sqrt{|h_k^{\mathrm{c}}(t)|^2\eta(t)}}{|h_k^{\mathrm{c}}(t)|^2+\eta(t)\lambda_k}\right)^2, \tag{12}$$

where $\eta(t)$ can be optimized by applying bisection search to solve the following problem

$$\sum_{k\in\mathcal{K}}\frac{\zeta_k(t)}{\left(\zeta_k(t)\nu(t)+1\right)^2}=\sigma^2 \tag{13}$$

with $\zeta_k(t)=\frac{|h_k^{\mathrm{c}}(t)|^2}{\lambda_k}$ and $\nu(t)=\frac{1}{\eta(t)}$. The dual variables $\{\lambda_k\}_{k\in\mathcal{K}}$ can be optimized via a sub-gradient based method. Specifically, to obtain the optimal $\{\lambda_k\}_{k\in\mathcal{K}}$ that maximize $G(\{\lambda_k\})$, the dual variables $\{\lambda_k\}_{k\in\mathcal{K}}$ can be updated as follows

$$\lambda_k\leftarrow\lambda_k+m\left(\frac{1}{T}\sum_{t=1}^{T}p_k(t)-\bar{P}_k\right),\ \forall\, k\in\mathcal{K}, \tag{14}$$

where $\frac{1}{T}\sum_{t=1}^{T}p_k(t)-\bar{P}_k$ is the sub-gradient of $G(\{\lambda_k\})$, $\forall\, k\in\mathcal{K}$ and m is the step size.

B. RIS Phase-Shift Optimization

For a given $\boldsymbol{p}(t)$ and $\eta(t)$, problem (11) can be rewritten as

$$\begin{aligned}\mathscr{P}_1: \ &\min_{\boldsymbol{v}(t)}: \phi(\boldsymbol{v}(t))\\ &\text{s.t. } |\boldsymbol{v}_j(t)|=1, \forall\, j=1,\ldots,N, \ \ \forall\, t=1,\ldots,T,\end{aligned}$$

where

$$\phi(\boldsymbol{v}(t)) = \sum_{k=1}^{K} \Bigg[\frac{p_k(t)|h_k^{\mathrm{d}}(t) + \boldsymbol{a}_k(t)\boldsymbol{v}(t)|^2}{\eta} - 2\sqrt{\frac{p_k(t)}{\eta(t)}}|h_k^{\mathrm{d}}(t) + \boldsymbol{a}_k(t)\boldsymbol{v}(t)| \Bigg],$$

the constant term $\sum_{k\in\mathcal{K}} \lambda_k p_k(t)$ is ignored, and $\boldsymbol{a}_k(t) = \boldsymbol{g}(t)\mathrm{diag}(\boldsymbol{h}_k^{\mathrm{r}}(t))$. By defining $\check{\boldsymbol{v}}(t) = [\boldsymbol{v}(t), 1]^T$, $\boldsymbol{r}_k(t) = [\boldsymbol{a}_k(t), h_k^{\mathrm{d}}(t)]$, and matrices $\boldsymbol{R}_k(t) = \boldsymbol{r}_k^{\mathrm{H}}(t)\boldsymbol{r}_k(t)$, problem $\mathscr{P}_1$ is given by

$$\begin{aligned} \mathscr{P}_2 \ : \ & \min_{\check{\boldsymbol{v}}(t)} \sum_{k=1}^{K} \left(\sqrt{\frac{p_k(t)}{\eta(t)} \check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)} - 1 \right)^2 \\ & \text{s.t. } |\boldsymbol{v}_j(t)| = 1, \forall\, j = 1, \ldots, N, \end{aligned} \tag{15}$$

which is non-convex due to the square root operation. Note that the optimal $\check{\boldsymbol{v}}$ for problem $\mathscr{P}_2$ forces $\sqrt{\frac{p_k(t)}{\eta(t)} \check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)}$ to approach 1. Hence, we can use a convex function with respect to $\check{\boldsymbol{v}}$, i.e., $\frac{p_k(t)}{\eta(t)} \check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)$, to approximate the square root term. By replacing $\sqrt{\frac{p_k(t)}{\eta(t)} \check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)}$ with $\frac{p_k(t)}{\eta(t)} \check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)$, problem $\mathscr{P}_2$ can be converted to the following non-convex quadratically constrained quadratic programming (QCQP) problem

$$\begin{aligned} \mathscr{P}_3 \ : \ & \min_{\check{\boldsymbol{v}}(t)} \sum_{k=1}^{K} \left(\frac{p_k}{\eta} \check{\boldsymbol{v}}^{\mathrm{H}} \boldsymbol{R}_k \check{\boldsymbol{v}} - 1 \right)^2 \\ & \text{s.t. } |\check{\boldsymbol{v}}_j(t)|^2 = 1, \ \ \forall\, j = 1, \ldots, N, \ \ \forall\, t = 1, \ldots, T, \end{aligned} \tag{16}$$

where $\check{\boldsymbol{v}}_n(t)$ is the n-th element of $\check{\boldsymbol{v}}(t)$. To convexify problem $\mathscr{P}_3$, we adopt matrix lifting to linearize problem $\mathscr{P}_3$. We denote $\check{\boldsymbol{v}}^{\mathrm{H}}(t)\boldsymbol{R}_k(t)\check{\boldsymbol{v}}(t)$ as $\mathrm{Tr}(\boldsymbol{R}_k(t)\boldsymbol{V}(t))$, where $\boldsymbol{V}(t) = \check{\boldsymbol{v}}(t)\check{\boldsymbol{v}}^{\mathrm{H}}(t)$. Problem $\mathscr{P}_3$ can be rewritten as

$$\begin{aligned} \mathscr{P}_4 \ : \ & \min_{\boldsymbol{V}} \sum_{k=1}^{K} \left(\frac{p_k(t)}{\eta(t)} \mathrm{Tr}(\boldsymbol{R}_k(t)\boldsymbol{V}(t)) - 1 \right)^2 \\ & \text{s.t. } \boldsymbol{V}_{i,i}(t) = 1, \ \ \forall i \ = 1, \ldots, N+1, \\ & \quad \mathrm{rank}(\boldsymbol{V}(t)) = 1, \\ & \quad \boldsymbol{V}(t) \succeq 0, \ \ \forall\, t = 1, \ldots, T. \end{aligned} \tag{17}$$

For the positive semi-definite (PSD) matrix $\boldsymbol{V}$, constraint $\mathrm{rank}\,(\boldsymbol{V}(t)) = 1$ is equivalent to $\mathrm{Tr}(\boldsymbol{V}(t)) - \|\boldsymbol{V}(t)\|_2 = 0$, where $\|\boldsymbol{V}(t)\|_2$ denotes the spectral norm of matrix $\boldsymbol{V}(t)$. Hence, we reformulate problem $\mathscr{P}_4$ as

$$\begin{aligned} \mathscr{P}_5 \ : \ & \min_{\boldsymbol{V}(t)} \sum_{k=1}^{K} \left(\frac{p_k(t)}{\eta(t)} \mathrm{Tr}\left(\boldsymbol{R}_k(t)\boldsymbol{V}(t)\right) - 1 \right)^2 + \mu(t) \\ & \text{s.t. } \boldsymbol{V}_{i,i}(t) = 1, \ \forall\ i \ = 1, \ldots, N+1, \\ & \quad \boldsymbol{V}(t) \succeq 0, \ \ \forall\, t = 1, \ldots, T, \end{aligned} \tag{18}$$

where $\mu(t) = \rho(\mathrm{Tr}(\boldsymbol{V}(t)) - \|\boldsymbol{V}(t)\|_2)$. Problem $\mathscr{P}_5$ is a semi-definite programming (SDP) problem. By forcing the difference between trace norm and spectral norm to be zero, an exact rank-one optimal solution $\boldsymbol{V}^*(t)$ can be found. By decomposing via Cholesky decomposition, i.e., $\boldsymbol{V}^*(t) = \check{\boldsymbol{v}}^{\mathrm{H}}(t)\check{\boldsymbol{v}}(t)$, we obtain a feasible solution $\check{\boldsymbol{v}}(t)$. If the objective value of (18) fails to be zero, then problem $\mathscr{P}_5$ is considered to be infeasible.

By now, we propose an alternating optimization algorithm to solve problem $\mathscr{P}_0$. In each iteration, the phase-shift vector is optimized by solving the convex problem $\mathscr{P}_5$. After updating the combined channel coefficients with optimized $\boldsymbol{v}^*(t)$, we can update $\boldsymbol{p}^*(t)$, $\eta^*(t)$, and $\{\lambda_k^*\}_{k\in\mathcal{K}}$ in an alternating manner by utilizing (12), (13), and (14), respectively.

Though the semi-definite relaxation (SDR) based method can be applied to solve problem (17) by ignoring the rank-one constraint, the returned solution for such a relaxed SDP problem may fail in meeting the rank-one constraint, where the Gaussian randomization method can then be adopted to obtain a suboptimal solution. In contrast, the proposed optimization-based method represents the rank-one constraint with the difference between the trace norm and spectral norm being zero. This mitigates the drawback of the SDR-based method, in particular when the number of RIS reflective elements is large. In addition, though the proposed alternating optimization algorithm is effective in solving problem $\mathscr{P}_0$, it is computationally expensive due to the following reasons. First, the optimization of $\eta(t)$ is achieved via bisection search for the solution of (13), which is required in each inner iteration. Second, phase-shift vector $\boldsymbol{v}(t)$ is optimized by solving several SDP problems, and the computational complexity in solving a single SDP problem increases exponentially with the number of RIS elements. Third, the optimization-based algorithm is executed in an alternating manner, which further increases the computation complexity. From the perspective of aggregation accuracy, the optimization-based algorithm only achieves sub-optimal performance because of the alternating optimization.

C. Computation Complexity

We tackle problem $\mathscr{P}_0$ by solving a series of SDP problems. The computational complexity of solving each SDP problem is $\mathcal{O}((N+1)^{4.5}\log(1/\epsilon_2^{\mathrm{bis}}))$, where $\epsilon_2^{\mathrm{bis}}$ denotes the accuracy. Therefore, the computational complexity for RIS phase-shifts optimization is $\mathcal{O}(J(N+1)^{4.5}\log(1/\epsilon_2^{\mathrm{bis}}))$, where J is the number of the SDR problems. For optimizing transmit power and denoising factor, the main iteration is bisection search for optimal η, thereby the computational complexity is $\mathcal{O}(\log(1/\epsilon_1^{\mathrm{bis}}))$, where $\epsilon_1^{\mathrm{bis}}$ is the accuracy of bisection search. Hence, the computational complexity is $\mathcal{O}(G(J(N+1)^{4.5}\log(1/\epsilon_2^{\mathrm{bis}}) + \log(1/\epsilon_1^{\mathrm{bis}})))$, where G is the number of alternating iterations. The computational complexity is dominated by the optimization of RIS, the complexity of which grows exponentially with the number of RIS elements.

To this end, we shall develop a GNN-based learning algorithm, which is of low computation complexity and achieves joint optimization for RIS-assisted over-the-air FL.

V. GNN-based Learning Algorithm

In this section, we develop a novel GNN-based learning algorithm to solve problem $\mathscr{P}_0$, and elaborate on the corresponding architecture design and training of neural networks.

A. Graphical Representation

To circumvent the limitations of the optimization-based algorithm, we develop a GNN-based learning framework to learn a direct mapping between the channel coefficients and the optimal parameter setting. We denote the mapping function by $\kappa(\cdot)$, which maps the channel coefficients (i.e., $\{h_i^{\mathrm{d}}(t)\}$ and $\{\boldsymbol{g}(t)\mathrm{diag}(\boldsymbol{h}_i^{\mathrm{r}}(t))\}$) to devices' transmit powers (i.e., $\boldsymbol{p}(t)$), denoising factor (i.e., $\eta(t)$), and RIS phase-shift vector (i.e., $\boldsymbol{v}(t)$). Hence, we express the mapping function as

$$\{\boldsymbol{p}(t), \boldsymbol{v}(t), \eta(t)\} = \kappa(h_i^{\mathrm{d}}(t),\ \boldsymbol{g}(t)\mathrm{diag}(\boldsymbol{h}_i^{\mathrm{r}}(t))), \tag{19}$$

and solving problem $\mathscr{P}_0$ can be interpreted as learning the optimal mapping function $\kappa(\cdot)$. We leverage the universal approximation property of deep neural networks (DNN) to parameterize the mapping function $\kappa(\cdot)$ between the channel coefficients and the optimized parameter setting, and then train the neural networks in a data-driven manner to learn the optimal mapping function.

As the objective of problem $\mathscr{P}_0$ is to simultaneously minimize the error due to signal misalignment and the receiver noise, the AirComp transceiver design (i.e., transmit power and denoising factor) and the RIS phase-shifts should be optimized in a coordinated manner. Hence, we develop a GNN-based learning framework to learn the mapping function and enhance the learning performance. The graph consists of $K+2$ nodes, which are connected by $2K+1$ edges. In particular, the K edge devices are represented by nodes 1 to K, while the edge server and the RIS are represented by node $K+1$ and node $K+2$, respectively. The representation vector of node k is denoted as $\boldsymbol{z}_k$, which should be trained to have all the useful information required for establishing the mapping. The representation vectors are updated in a layer-wise manner, where each layer consists of both combining and aggregation operations that take the representation vectors in the preceding layer as the input and will be elaborated in the next subsection. After updating, all representation vectors should have enough information for the joint AirComp transceiver and RIS phase-shifts design. We can then directly obtain the optimized setting of AirComp transceiver from representation vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{K+1}$ and that of RIS from representation vector $\boldsymbol{z}_{K+2}$.

Compared with conventional fully connected neural networks (FCNN), GNN has the following advantages. First, GNN can capture the interaction among multiple edge devices, RIS, and edge server via the combining and aggregation operations, i.e., the update of each node exploits the representation vectors of other nodes. Hence, the coupling among the optimization variables is captured by GNN to achieve joint optimization for aligning the signals transmitted by different edge devices and meanwhile reducing the detrimental impact of receiver noise, thereby enhancing the learning performance. Second, the inherent permutation equivalence and permutation invariance properties of GNN can be leveraged to enhance the scalability of the developed learning framework [46]. In particular, permutation equivalence ensures that a permutation of device channels leads to the same permutation of the transmit power control vector $\boldsymbol{p}(t)$, while the permutation invariance ensures that the phase-shift vector and the denoising factor is independent of the permutation of device channels. Third, in each layer of GNN, all edge devices share the same modules for combining and learning operations. Meanwhile, GNN is trained by optimizing the parameters of these modules in different layers, which is very different from optimizing the weights between adjacent hidden layers in FCNN. Hence, GNN reduces the model complexity as its parameter dimension does not depend on the number of edge devices. As a result, the proposed GNN-based learning framework is scalable. When the amount of edge devices changes, the proposed framework can adapt to the new scenario by simply adjusting the number of modules, while the FCNN-based framework has to re-train the neural network. In summary, the proposed framework is of low model complexity and high training efficiency and has the features of generalizability and scalability.

B. GNN Architecture Design

In this subsection, we describe the architecture of the proposed GNN-based learning algorithm, which is deployed and trained at the edge server. By feeding the channel coefficients into a well-trained GNN, the edge server is capable of obtaining the optimized AirComp transceiver and RIS phase-shifts design. Subsequently, the optimized design parameters are fed back to the edge devices and RIS controller, and then used for uplink model aggregation. The proposed GNN-based learning algorithm consists of an initialization layer, multiple graphical mapping layers, and one parameter generation layer. It first transforms the channel coefficients to the representation vectors $\boldsymbol{z}_k^{(0)}, \forall\ k = 1, \ldots, K+2$ through the initialization layer, then updates the representation vectors of all nodes via D graphical mapping layers to obtain $\boldsymbol{z}_k^{(D)}, \forall\ k = 1, \ldots, K+2$, and finally obtains the learned parameters of the AirComp transceiver (i.e., $\{\eta(t), \boldsymbol{p}(t)\}$) and the RIS phase-shifts (i.e., $\boldsymbol{v}(t)$) via the parameter generation layer. The overall network structure is illustrated in Fig. 2. We elaborate on the design of all these layers as follows:

- *Initialization layer:* The initialization layer is designed to transform channel coefficients $\{h_k^{\mathrm{d}}(t), \boldsymbol{g}(t)\boldsymbol{h}_k^{\mathrm{r}}(t)\}_{k\in\mathcal{K}}$ to representation vectors $z_k^{(0)}, \forall\ k = 1, \ldots, K+2$. The representation vector of each device node is initialized by passing its channel coefficients through a multi-layer perceptron (MLP) based encoder, denoted by $f_{\mathrm{EC}}^0(\cdot)$. This encoder is composed of three linear layers, where a batch normalization layer and an activation layer are placed sequentially between two adjacent linear layers. We adopt the Rectified Linear Unit (ReLU) as the activation function due to its linear mapping property and low complexity in the back-propagation calculation. As the channel coefficients are complex-valued that cannot be fully supported by the current deep learning toolkits, we separate the real and imaginary components of channel coefficients $\{h_k^{\mathrm{d}}(t), \boldsymbol{g}(t)\boldsymbol{h}_k^{\mathrm{r}}(t)\}$ and feed them into encoder $f_{\mathrm{EC}}^0(\cdot)$. Hence, the representation vector

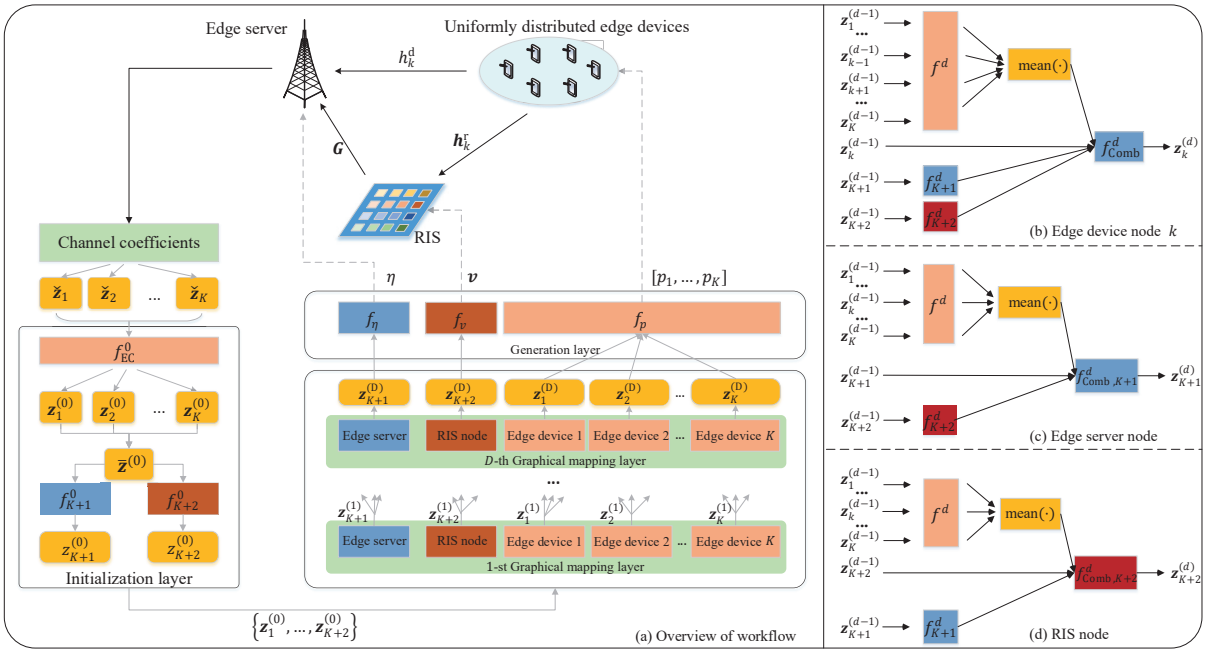


Fig. 2. (a) Overall workflow of the proposed GNN-based learning algorithm for RIS-assisted over-the-air FL; (b) Module design of the d-th graphical mapping layer for edge device nodes; (c) Module design of the d-th graphical mapping layer for edge server node; (d) Module design of the d-th graphical mapping layer for edge server node.

$\boldsymbol{z}_k^{(0)}$ of edge device node $k \in \mathcal{K}$ can be expressed as

$$\boldsymbol{z}_k^{(0)} = f_{\text{EC}}^0([\Re(h_k^{\text{d}}(t) + \boldsymbol{g}(t)\boldsymbol{h}_k^{\text{r}}(t)), \Im(h_k^{\text{d}}(t) + \boldsymbol{g}(t)\boldsymbol{h}_k^{\text{r}}(t))]).$$

Note that all edge device nodes share the same encoder to ensure permutation equivalence. This also reduces the number of parameters required to be trained as well as the computational complexity in forward inference. By denoting the average of the representation vectors of all edge device nodes as $\bar{\boldsymbol{z}}^{(0)} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{z}_k^{(0)}$, the representation vectors of the edge server and the RIS, denoted as $f_{K+1}^0(\cdot)$ and $f_{K+2}^0(\cdot)$, respectively, can be written as

$$\boldsymbol{z}_{K+1}^{(0)} = f_{K+1}^0(\bar{\boldsymbol{z}}^{(0)}), \qquad \boldsymbol{z}_{K+2}^{(0)} = f_{K+2}^0(\bar{\boldsymbol{z}}^{(0)}). \quad (20)$$

The averaging operation is adopted to obtain the representation vectors of nodes $K+1$ and $K+2$, as the representation vectors of all device nodes affect that of the edge server and the RIS.

Remark 1. *Note that the FCNN initializes the representation vector by applying the matrix multiplication on a vector composed of all channel coefficients and the weights of neurons, where the matrix multiplication is sensitive to the order of the channel coefficients. In contrast, the proposed GNN architecture initializes the representation vector based on the individual channel coefficients, which ensures permutation invariance.*

- *Graphical mapping layer:* Each graphical mapping layer consists of $K+2$ modules for $K+2$ nodes. Each module executes the aggregation and combining operations based on the representation vectors of all nodes from the preceding layer. In general, the update of each node in the d-th graphical mapping layer is conducted as in [47]

$$\boldsymbol{z}_k^{(d)} = f_{\text{comb},k}^d\left(\boldsymbol{z}_k^{(d-1)}, f_{\text{agg}}^d(\{\boldsymbol{z}_j^{(d-1)}\}_{j\neq k})\right), \quad (21)$$

where $f_{\text{comb},k}^d(\cdot)$ and $f_{\text{agg}}^d(\cdot)$ denote the combining function and the aggregation function at layer d, respectively. In particular, the aggregation function of node k is designed to aggregate the representation vectors $\boldsymbol{z}_j^{(d-1)}$ from all other nodes, while the combining function is adopted to combine the representation vector of local node $\boldsymbol{z}_k^{(d-1)}$ with the aggregated representation vectors $\{\boldsymbol{z}_j^{(d-1)}\}_{j\neq k}$ to update the local representation vector $\boldsymbol{z}_k^{(d)}$. We follow the design in [34], [46] and approximate functions $f_{\text{agg}}(\cdot)$ as follows

$$f_{\text{agg}}^d\left(\{\boldsymbol{z}_j^{(d-1)}\}_{j\neq k}\right) = \phi\left(\{f^d(\boldsymbol{z}_j^{(d-1)})\}_{j\neq k}\right),$$

where $\phi(\cdot)$ is a function that keeps permutation invariance, e.g., element-wise mean pooling and $f^d(\cdot)$ denotes an MLP to encode the representation vector. Besides, $f_{\text{comb},k}^d(\cdot)$ is parameterized by an MLP encoder. For all the modules in the same layer, the same encoder is adopted to encode the representation vector of each node. This ensures the permutation equivalence of each device node and also reduces the number of parameters for improving the robustness of the neural network. The detailed designs of different modules are different because of the difference in aggregation and combining operations among the nodes.

Each device node $k \in \mathcal{K}$ aggregates the information from all other nodes, including device nodes, RIS node, and edge server node. This guarantees that sufficient CSI can be acquired by each node, and the coordination among the edge devices, the RIS, and the edge server can be achieved. Hence, the representation vector at node k is updated as follows

$$\boldsymbol{z}_k^{(d)} = f_{\text{comb},k}^d \Big(\boldsymbol{z}_k^{(d-1)}, \boldsymbol{z}_{\text{agg},k}^{(d)}, f_{K+1}^d(\boldsymbol{z}_{K+1}^{(d-1)}), f_{K+2}^d(\boldsymbol{z}_{K+2}^{(d-1)}) \Big), \tag{22}$$

where

$$\boldsymbol{z}_{\text{agg},k}^{(d)} = \frac{1}{K-1} \sum_{j \neq k}^{1 \le j \le K} \left(f^d(\boldsymbol{z}_j^{(d-1)}) \right), \tag{23}$$

$f_{K+1}^d(\cdot)$ and $f_{K+2}^d(\cdot)$ denote the encoders specifically designed for node $K+1$ and node $K+2$, respectively. The update process is illustrated in Fig. 2. Note that we treat the representation vectors of other device nodes (i.e., $\boldsymbol{z}_j^{(d-1)}$) and that of the edge server and the RIS separately in (22). Specifically, node $k \in \mathcal{K}$ aggregates the average of the encoded representation vector of other nodes, because the signal alignment depends on all channels instead of the strongest one. Meanwhile, node $k \in \mathcal{K}$ aggregates the representation vectors of the RIS node and the edge server node, which do not change their permutation invariance property. Such a design also enables the neural network to better learn the channel representation with respect to its channel information at the RIS node and at the edge server node. The update of the representation vectors of the RIS node and the edge server node can be expressed as $\boldsymbol{z}_{K+1}^{(d)} = f_{\text{Comb},K+1}^d \left(\boldsymbol{z}_{\text{agg}}^{(d)}, f_{K+2}^d(\boldsymbol{z}_{K+2}^{(d-1)}), \boldsymbol{z}_{K+1}^{(d-1)} \right)$ and $\boldsymbol{z}_{K+2}^{(d)} = f_{\text{Comb},K+2}^d \left(\boldsymbol{z}_{\text{agg}}^{(d)}, f_{K+1}^d(\boldsymbol{z}_{K+1}^{(d-1)}), \boldsymbol{z}_{K+2}^{(d-1)} \right)$, respectively, where

$$\left[\boldsymbol{z}_{\text{agg}}^{(d)} \right]_j = \frac{1}{K} \sum_{i=1}^{K} \left[f^d(\boldsymbol{z}_i^{(d-1)}) \right]_j, \forall\, j = 1, \ldots, l(d). \tag{24}$$

The update flows are illustrated in Fig. 2, and Fig. 3 illustrates how the RIS node updates its representation vector from two edge device nodes and one edge server node. Specifically, the RIS node first aggregates the encoded representation vectors $f^d(\boldsymbol{z}_1^{(d-1)})$, $f^d(\boldsymbol{z}_2^{(d-1)})$, and $f_{K+1}^d(\boldsymbol{z}_3^{(d-1)})$ from the edge device nodes and the edge server node. Then it obtains $\boldsymbol{z}_{\text{agg}}^{(d)}$ by applying the mean pooling to the representation vectors of the edge device nodes. By concatenating $\boldsymbol{z}_{\text{agg}}^{(d)}$, $f_{K+1}^d(\boldsymbol{z}_3^{(d-1)})$, and $\boldsymbol{z}_4^{(d-1)}$ and then passing them to function $f_{\text{comb},4}^d(\cdot)$, the RIS node obtains the updated representation vector $\boldsymbol{z}_4^{(d)}$. We adopt the element-wise mean pooling as the RIS and the edge server receive signals from all edge devices. Note that various functions can be selected to keep permutation equivalence and permutation invariance in the aggregation and combining operations. We shall show the excellent performance of the proposed design in Section VI.

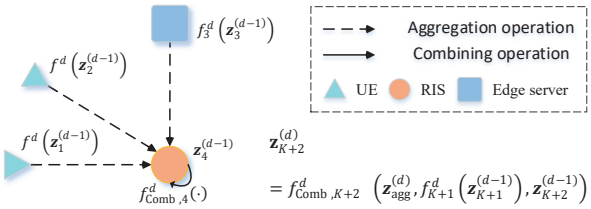


Fig. 3. An example of how the RIS node aggregates and combines information from edge device and edge server nodes.

- *Parameter generation layer:* Through the aggregation and combining over D graphical mapping layers, we obtain the representation vectors that have sufficient information. Then, the generation layer maps them into the desired transmit power, the phase-shift vector, and the denoising factor. In particular, we apply three decoders, denoted by f_p, f_v, and f_η, to decode the representation vectors of the edge server node, the RIS node, and the edge device nodes, respectively. Mathematically, we have $p_k = f_p(\boldsymbol{z}_k^{(d)}), \eta = f_\eta(\boldsymbol{z}_{K+1}^{(d)}), \boldsymbol{v} = f_v(\boldsymbol{z}_{K+2}^{(d)})$. Each decoder has the same design as the encoders in the initialization layer. We adopt the Sigmoid activation layer to bound the output range to be $[-1, 1]$. The corresponding design of each node can be scaled to the actual level by using an affine transformation.

Remark 2. *The learnable functions in the proposed GNN-based learning algorithms, i.e., $f_{Comb}^d(\cdot)$, $f_{Comb,K+1}^d(\cdot)$, $f_{Comb,K+2}^d(\cdot)$, $f^d(\cdot)$, $f_{K+1}^d(\cdot)$, $f_{K+2}^d(\cdot)$, $f_{EC}^0(\cdot)$, $f_p(\cdot)$, $f_v(\cdot)$, $f_\eta(\cdot), \forall\, d = 0, \ldots, D$, are independent of the number of edge devices. Hence, the proposed GNN-based learning algorithm is scalable and does not require re-training when the number of edge devices varies, which is a key advantage of GNN over the conventional FCNN.*

C. Loss Function Design and Training

The proposed learning algorithm is trained offline with a mini-batch of samples in an unsupervised manner, where the training data are uniformly sampled from the training set. To minimize the time-average error with average transmit power constraints, we design the loss function as

$$\text{loss} = \frac{1}{B} \sum_{m=1}^{B} \left[\text{Er}(t) + \sum_{k \in \mathcal{K}} \text{Reg}(p_k(t), \bar{p}_k) \right], \tag{25}$$

where B denotes the size of mini-batch and $\text{Reg}(p_k(t), \bar{p}_k)$ denotes the regularizer for the average power constraints. As the average power constraint is unidirectional, the ideal regularizer function $\psi(x, \bar{x})$ should be 0 if $x \le \bar{x}$, and $h(x)$ otherwise, where $h(x) > 0$ is a monotonic increasing function with respect to x. Hence, to reduce the computational complexity of back-propagation, we define $h(x) = x$ and $\psi(x, \bar{x}) = \text{ReLU}(x - \bar{x}) = \max(0, x - \bar{x})$. Thus, the variables

to be optimized are coupled in the loss function, and the average power constraints are considered.

The parameters of the neural network are optimized by adopting stochastic gradient descent according to the loss function. To avoid overfitting, we set an early stopping criterion to stop the training process. Specifically, the neural network is considered to converge when the variance of the sequence consisting of the last 100 logarithmic training losses is smaller than ϵ_{stop}.

D. Computation Complexity

We start by analyzing the computational complexity of the forward inference process. As the encoders/decoders across different layers share the same structure, the computational complexity of each encoder/decoder operation is considered to be the same and denoted as $\mathcal{O}(C)$. For the initialization layer, as there are $K+2$ encoding operations and one average operation based on K representation vectors, the computational complexity is $\mathcal{O}(C(K+2)+K)$. For the graphical mapping layers, each module performs the aggregation and combining operations. According to (23), the computational complexity of aggregation operations of the edge device node is $\mathcal{O}(C(K+1)+K-1)$. For the RIS and edge server nodes, according to (24), the computation complexity of aggregation operations is $\mathcal{O}(C(K+1)+K)$. The combining operations at all nodes are the same and the computational complexity is $\mathcal{O}(C)$. Thus, the computational complexity of D graphical mapping layers is $\mathcal{O}\left(D(K^2(C+1)+(4C+1)K+4C)\right)$. The computational complexity of the generation layer is $\mathcal{O}((K+2)C)$ due to the decoding operations. As a result, the overall computational complexity of the forward inference is $\mathcal{O}\left(DK^2(C+1)+(6C+2)K+8C\right) \approx \mathcal{O}\left(RDK^2+SK\right)$, where $R = C+1$, $S = 6C+2$, and the constant item is omitted. As the training process involves B samples, the computational complexity of the backward-propagation is $\mathcal{O}\left(BRDK^2+BSK\right)$, where the operation time for loss calculation is ignored. The computation complexity of the GNN-based learning framework is proportional to the edge device number. Parameters R and S are determined by the number of parameters of each encoder/decoder. Hence, adopting the encoders/decoders with a small number of neurons reduces the computational complexity. Moreover, the computational complexity increases linearly with D for a given number of edge devices.

VI. Simulation Results

A. Simulation Setup

We consider an RIS-assisted over-the-air FL in a three-dimensional coordinate system. An edge server and the RIS are deployed at $(-200, 10, 30)$ and $(0, 0, 10)$ meters, respectively, and the edge devices are uniformly distributed in the circular area centered at $(50, 0, 0)$ with a radius of 25 meters. The RIS reflection elements are arranged in a 10×10 pattern on the (y, z)-plane.

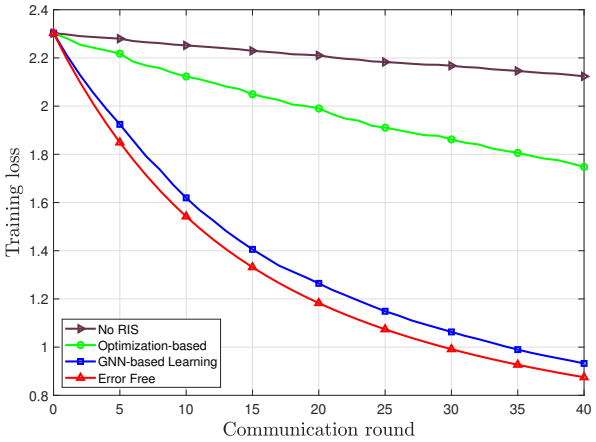
The channel response of the edge server-device k link is $h_k^{\text{d}}(t) = \gamma_k^{\text{d}} \tilde{h}_k^{\text{d}}(t)$, where $\gamma_k^{\text{d}} = \sqrt{D_0 d_{\text{DF},k}^{-\alpha_1}}$, $\tilde{h}_k^{\text{d}}(t) \sim \mathcal{CN}(0,1)$, $d_{\text{DF},k}$ is the distance between device k and the edge server, $\alpha_1 = 4$ is the direct link's path loss exponent, and $D_0 = -25$ dB denotes the path loss at the reference distance. The channel coefficients $\boldsymbol{g}(t)$ and $\boldsymbol{h}_i^{\text{r}}(t)$ follow the Rician distribution, i.e., $\boldsymbol{g}(t) = \gamma^{\text{r},1}\left(\sqrt{\frac{\delta}{\delta+1}}\tilde{\boldsymbol{g}}^{\text{LOS}}(t) + \sqrt{\frac{1}{1+\delta}}\tilde{\boldsymbol{g}}^{\text{NLOS}}(t)\right)$ and $\boldsymbol{h}_k^{\text{r}}(t) = \gamma_k^{\text{r},2}\left(\sqrt{\frac{\delta}{\delta+1}}\tilde{\boldsymbol{h}}_k^{\text{r,LOS}}(t) + \sqrt{\frac{1}{1+\delta}}\tilde{\boldsymbol{h}}_k^{\text{r,NLOS}}(t)\right)$, where LOS and NLOS represent the line-of-sight and non-line-of-sight paths, respectively, and $\delta = 10$ denotes the Rician factor. Similarly, $\gamma^{\text{r},1} = \sqrt{D_0 d_{\text{IF}}^{-\alpha_2}}$ and $\gamma_k^{\text{r},2} = \sqrt{D_0 d_{\text{DR},k}^{-\alpha_3}}$ denote the large-scale fading between the edge server and the RIS, and the RIS and device k, respectively, where $\alpha_2 = 1.8$ and $\alpha_3 = 2.1$. d_{IF} and $d_{\text{DR},k}$ are the distances between RIS and edge server, and RIS and device k, respectively. The non-line-of-sight paths, i.e., $\tilde{\boldsymbol{g}}^{\text{NLOS}}(t)$ and $\tilde{\boldsymbol{h}}_k^{\text{r,NLOS}}(t)$, obey the complex Gaussian distribution. For line-of-sight paths, by denoting the azimuth and elevation angles of departure (AOD) from the RIS to the edge server as θ_1^{A} and θ_1^{E}, respectively, we have $\tilde{\boldsymbol{g}}^{\text{LOS}}(t) = \boldsymbol{a}_{\text{RIS}}(\theta_1^{\text{A}}, \theta_1^{\text{E}}) a_{\text{ES}}$, where $\boldsymbol{a}_{\text{RIS}}(\theta_1^{\text{A}}, \theta_1^{\text{E}})$ denotes the steering vector of RIS, and a_{ES} denotes the steering scalar of edge server. In particular, $a_{\text{ES}} = 1$ as one antenna is considered at the edge server. The n-th element of $\boldsymbol{a}_{\text{RIS}}(\theta_1^{\text{A}}, \theta_1^{\text{E}})$ is $\left[\boldsymbol{a}_{\text{RIS}}(\theta_1^{\text{A}}, \theta_1^{\text{E}})\right]_n = e^{j\beta\frac{2\pi d_{\text{e}}}{\xi}\omega(\theta_1^{\text{A}}, \theta_1^{\text{E}}, n)}$, where $\omega(\theta_1^{\text{A}}, \theta_1^{\text{E}}, n) = \left\lfloor \frac{n-1}{10} \right\rfloor \sin(\theta_1^{\text{E}}) + \text{mod}(n-1, 10)\sin(\theta_1^{\text{A}})\cos(\theta_1^{\text{E}})$, d_{e} denotes the interval between two adjacent passive elements of the RIS, and ξ denotes the carrier wavelength. Without loss of generality, we set $\frac{2\pi d_{\text{e}}}{\xi} = 1$. With given locations of edge server $(x_{\text{ES}}, y_{\text{ES}}, z_{\text{ES}})$ and RIS $(x_{\text{RIS}}, y_{\text{RIS}}, z_{\text{RIS}})$, we have $\sin(\theta_1^{\text{A}})\cos(\theta_1^{\text{E}}) = \frac{y_{\text{ES}} - y_{\text{RIS}}}{d_{\text{IF}}}$ and $\sin(\theta_1^{\text{E}}) = \frac{z_{\text{ES}} - z_{\text{RIS}}}{d_{\text{IF}}}$. By denoting the angles of arrival (AOA) in the azimuth and elevation directions from device k to the RIS by $\theta_{2,k}^{\text{A}}$ and $\theta_{2,k}^{\text{E}}$, we have $\tilde{\boldsymbol{h}}_k^{\text{r,LOS}}(t) = \boldsymbol{a}_{\text{RIS}}(\theta_{2,k}^{\text{A}}, \theta_{2,k}^{\text{E}})$. With given location of device k (i.e., $(x_{\text{d},k}, y_{\text{d},k}, z_{\text{d},k})$), we have $\sin(\theta_{2,k}^{\text{A}})\cos(\theta_{2,k}^{\text{E}}) = \frac{y_{\text{d},k} - y_{\text{RIS}}}{d_{\text{DR}}}$ and $\sin(\theta_{2,k}^{\text{E}}) = \frac{z_{\text{d},k} - z_{\text{RIS}}}{d_{\text{DR}}}$. The noise power is -75 dBm.

The proposed GNN-based learning algorithm is implemented using PyTorch with Adam optimizer[2]. We set the sizes of the training and testing datasets as 20000 and 5000, respectively. To validate the robustness of the proposed algorithm, we generate different sets of device locations for the training and testing data sets. The learning rate and the size of the mini-batch (i.e., B) are 0.001 and 1024, respectively. The proposed alternating optimization algorithm terminates when the decrease of the average MSE between two successive iterations is below 5×10^{-4}.

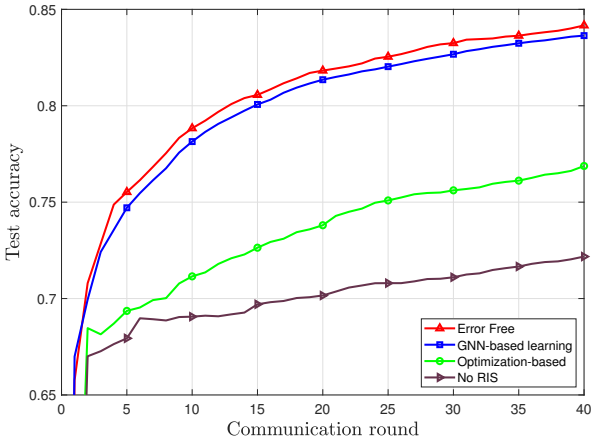
Benchmark Algorithms: We compare the proposed algorithms with two benchmarks, i.e., Error free and No RIS. For the Error free scheme, we assume that the uplink transmission of the local gradients is error-free. This benchmark represents the upper bound of the FL learning performance. For the No-RIS scheme, we alternately optimize the transmit power at the edge devices and the denoising factor at the edge server by adopting the method proposed in [38]. Specifically, by formulating the joint optimization problem as an unconstrained Lagrangian optimization problem, the transmit power, denois-

[2]The code for this paper can be found at https://github.com/XiaoWangya/GNNforOTAFL.git.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2023.3239400

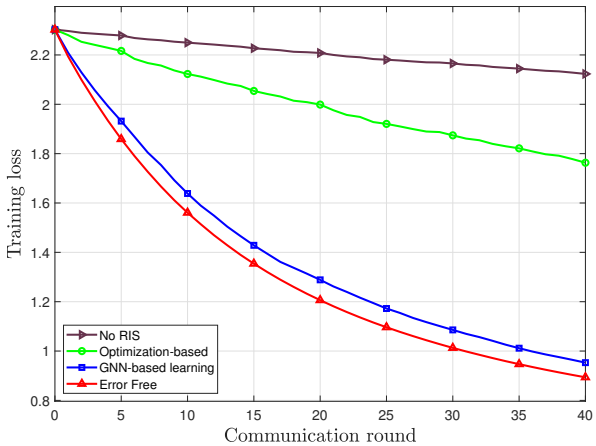


(a) Training loss

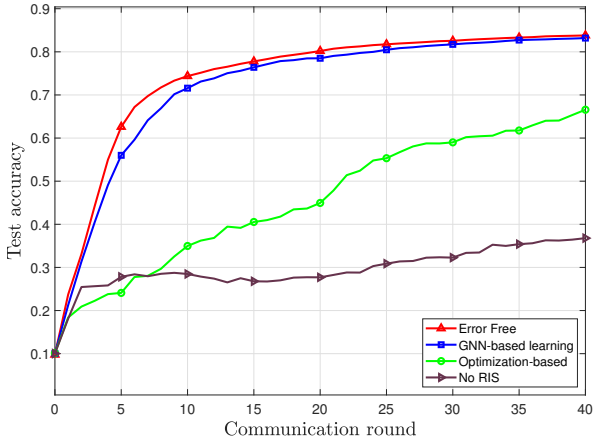


(b) Test accuracy

Fig. 4. Training loss and test accuracy versus number of communication rounds.



(a) Training loss



(b) Test accuracy

Fig. 5. Training loss and test accuracy versus number of communication rounds with non-i.i.d. data.

ing factor, and Lagrangian variables are optimized by applying the KKT condition, bisection search, and sub-gradient method, respectively.

FL Dataset: We evaluate the above algorithms with the handwritten digit recognition task based on the MNIST dataset. Each edge device is assigned the same amount of data labeled with $0-9$. The neural network that performs the classification task is an FCNN with three linear layers, where the activation function between adjacent linear layers is the Sigmoid function. We adopt the cross-entropy loss as the loss function.

B. Performance Comparison

Fig. 4 compares the training loss and test accuracy for the different amount of communication rounds. In Fig. 4(a), the training losses of all schemes decrease with the number of communication rounds. In particular, after 40 rounds of training, the proposed GNN-based learning algorithm achieves a much lower training loss than the optimization-based algorithm due to the following reasons. On one hand, the proposed GNN-based learning algorithm jointly optimizes the design of the AirComp transceiver and RIS phase-shifts and is trained with abundant samples in an unsupervised manner. On the other hand, the optimization-based algorithm achieves a sub-optimal solution because of the alternating operation and convex relaxation. Besides, the optimization-based algorithm achieves a better performance than the scheme without RIS, which shows the effectiveness of RIS in enhancing learning performance.

We illustrate the test accuracy of all the considered algorithms in Fig. 4(b). As the parameters of neural networks are yet to be trained, the aggregation error may enhance the robustness of the global model and result in a good performance in the earlier training. Hence, in the initial rounds, the optimization-based algorithm achieves the highest test accuracy. When the global model converges, the aggregation error is detrimental to improving the test accuracy. As the test accuracy is obtained by averaging over different sets of device locations and the same edge server and RIS locations, the permutation equivalence for both the edge server node and RIS node is guaranteed. The proposed GNN-based learning algorithm achieves a close test accuracy to the Error Free scheme and outperforms the optimization-based and the No RIS schemes.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2023.3239400

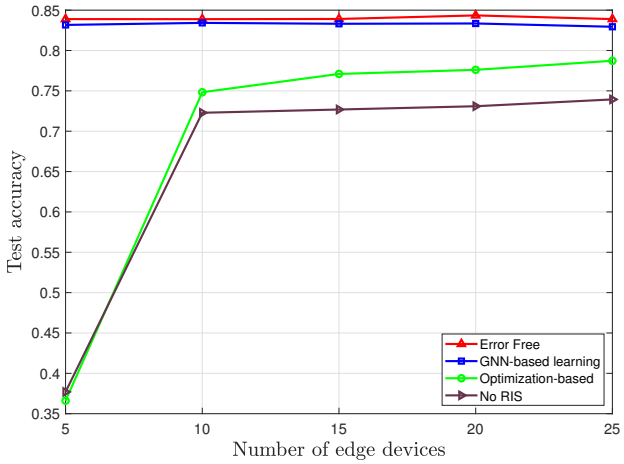
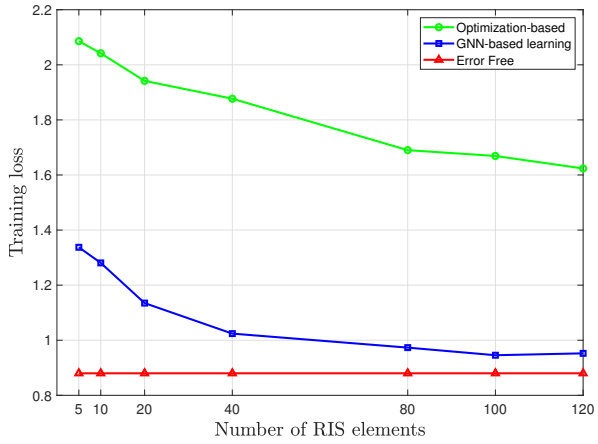


Fig. 6. Test accuracy versus number of edge devices.

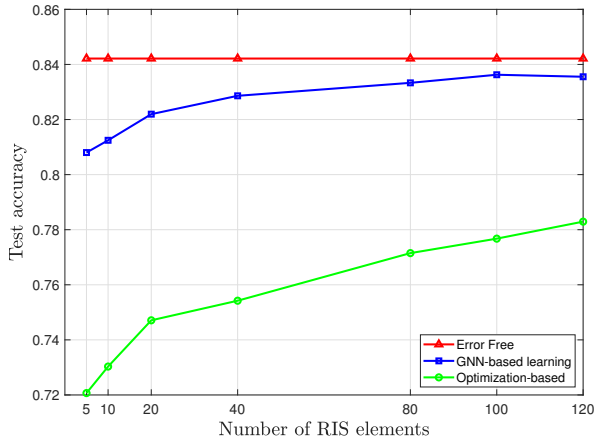
Fig. 5 illustrates the impact of non-i.i.d. data on both the convergence rate and test accuracy of the proposed GNN-based learning algorithm. Each edge device is assigned the same amount of feature-label pairs that include two random categories. Compared with the i.i.d. case, the proposed GNN-based learning algorithm in the non-i.i.d. case only suffers from a slight performance reduction due to the following reason. With non-i.i.d. data, the deviation of the direction of the global gradient increases with the number of local updates. As we consider one local update, such a deviation due to non-i.i.d. data is not significant. By balancing the tradeoff between signal-misalignment error and noise-induced error, our proposed GNN-based learning algorithm achieves a high aggregation accuracy, which is close to that of the Error Free scheme. Due to the insufficient exploitation of RIS, the optimization-based algorithm and the No RIS scheme achieve a high time-average error and suffer from high performance reduction.

We show the test accuracy of all algorithms under consideration for different number of edge devices in Fig. 6. According to (7), a greater number of edge devices tightens the upper bound in Theorem 1, thereby a higher aggregation accuracy and better FL performance can be achieved. The proposed GNN-based learning algorithm achieves a much higher test accuracy than other benchmarks, which demonstrates that the proposed GNN-based learning algorithm can effectively coordinate the AirComp transceiver and RIS. When $K = 5$, the test accuracy of both the optimization-based method and the No RIS scheme are low. This is because, when K is small, the optimal η shall be larger than 1×10^6, and the optimized transmit power is close to zero. As a result, the achieved aggregation error is high, leading to poor learning performance.

Fig. 7 shows the training loss and test accuracy under different number of RIS reflection elements. A greater number of RIS reflection elements leads to better channel qualities, which improves aggregation accuracy and achieves a better learning performance. By increasing the number of reflection elements from 5 to 120, the learning performance of the proposed GNN-based learning algorithm gradually gets closer



(a) Training loss



(b) Test accuracy

Fig. 7. Training loss and test accuracy versus number of reflection elements.

to the Error Free scheme, which shows the effectiveness of RIS. The residual gap is due to the initial optimality gap and gradient variance induced gap defined in Theorem 1. In addition, the proposed GNN-based learning algorithm outperforms the optimization-based algorithm, which demonstrates that the proposed GNN-based learning algorithm can make more efficient use of RIS to support the model aggregation. As the number of reflection elements increases, the increasing rate of the learning performance of the proposed GNN-based learning algorithm slows down.

We compare the running time of the proposed GNN-based learning algorithm and the optimization-based algorithm on an AMD EPYC 7742 platform with a GeForce RTX 3090. Since the existing python packages do not support full functional CVX programming, we conduct the optimization-based algorithm and No RIS scheme with MatLab. As shown in Table II, the proposed GNN can be trained offline using a high-performance computer and the training time for 5000 rounds is acceptable. Once the training is done, the proposed GNN can be directly applied to design AirComp transceiver and RIS phase-shifts. We observe that the average computation time of the proposed GNN-based learning algorithm is significantly

TABLE II
COMPUTATION TIME (SEC.) AND SOLUTION FEASIBILITY VERSUS NUMBER OF DEVICES

K	GNN-based		No RIS	Optimization-based	Feasibility ratio
	training (5000 rounds)	testing (per round)	testing (per round)	testing (per round)	
10	46.90	6.58×10^{-4}	0.018	15.06	100%
15	172.52	1.22×10^{-3}	0.019	20.34	100%
20	140.34	1.80×10^{-3}	0.019	25.88	100%
25	245.25	2.33×10^{-3}	0.020	28.25	100%
30	350.21	2.97×10^{-3}	0.03	31.84	100%

shorter than that of both the optimization-based algorithm and the No RIS scheme. When the number of devices is 10, the proposed GNN-based learning algorithm is about 2.29×10^4 and 27.35 times faster than the optimization-based algorithm and No RIS scheme, respectively. When the number of devices is 30, the advantage of the proposed GNN-based learning algorithm in terms of the computational complexity becomes more obvious, i.e., achieving 9.16×10^3 times speedup. The optimization-based algorithm suffers from higher computation complexity than the No RIS scheme, which shows that the iterative optimization of RIS phase-shifts is computationally expensive.

Table II also shows the feasibility ratio, which is defined as the percentage of the solutions generated by the proposed GNN satisfying the average power constraint over the test datasets. To eliminate potential randomness, we average the simulation results over 100 independent simulations, each of which includes 1000 communication rounds. As can be observed, the feasibility of the proposed GNN-based learning algorithm is always guaranteed for different number of devices. This is because incorporating the average transmit power constraints into the design of the loss function forces the proposed learning algorithm to meet those constraints.

VII. CONCLUSIONS

In this paper, we studied the joint design of an RIS-assisted over-the-air FL system. We theoretically derived the convergence upper bound of the proposed RIS-assisted over-the-air FL and formulated a joint optimization problem with respect to the transmit power, denoising factor, as well as RIS phase-shifts. To reduce the computation complexity and enhance the learning performance, we developed a GNN-based learning algorithm to solve the time-average error minimization problem. Extensive simulations showed the superiority of the proposed GNN-based learning algorithm in optimizing AirComp transceiver and phase-shifts in terms of low-complexity, high training efficiency, and scalability.

APPENDIX

A. Proof of Lemma 1

As in (4), we have $\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2] = \mathbb{E}\left[\left\|\frac{1}{K}\pi(t)\left(\hat{\boldsymbol{s}}(t) - \boldsymbol{s}(t)\right)\right\|_2^2\right]$. As there exists a constant $\Gamma \geq 0$ that upper bounds the variance of Ω elements of $\boldsymbol{\Upsilon}_k$ according to Assumption 4, we have $\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2] \leq \frac{\Gamma}{K^2}\mathbb{E}\left[\left\|\left(\hat{\boldsymbol{s}}(t) - \boldsymbol{s}(t)\right)\right\|_2^2\right]$. With the definition in (5), we have

$$\begin{aligned}&\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2]\\ \leq&\frac{\Gamma}{K^2}\mathbb{E}\left[\left\|\sum_{k=1}^{K}\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}\boldsymbol{I} - \boldsymbol{I}\right)\boldsymbol{s}_k(t) + \frac{\boldsymbol{n}(t)}{\sqrt{\eta(t)}}\right\|_2^2\right],\end{aligned}$$

which can be further decomposed according to Cauchy-Schwarz inequality into

$$\begin{aligned}\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2] \leq&\frac{\Gamma(K+1)}{K^2}\left\{\mathbb{E}\left[\left\|\frac{\boldsymbol{n}(t)}{\sqrt{\eta(t)}}\right\|_2^2\right] + \right.\\ &\left.\sum_{k=1}^{K}\mathbb{E}\left[\left\|\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}}\boldsymbol{I} - \boldsymbol{I}\right)\boldsymbol{s}_k(t)\right\|_2^2\right]\right\}.\end{aligned}$$

As the mean and variance of $\boldsymbol{s}_k(t)$ are zero and one, respectively, we have

$$\begin{aligned}&\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2]\\ =&\Omega\frac{\Gamma(K+1)}{K^2}\left\{\sum_{k=1}^{K}\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}} - 1\right)^2 + \frac{\sigma^2}{\eta(t)}\right\}\\ =&\Omega\frac{\Gamma(K+1)}{K^2}\mathrm{Er}(t),\end{aligned}$$

where $\mathrm{Er}(t) = \sum_{k=1}^{K}\left(\frac{\sqrt{p_k(t)}|h_k^{\mathrm{c}}(t)|}{\sqrt{\eta(t)}} - 1\right)^2 + \frac{\sigma^2}{\eta(t)}$.

B. Proof of Theorem 1

As $F_k(\boldsymbol{w})$ is S-smooth, $F(\boldsymbol{w})$ is also S-smooth, and we have

$$\begin{aligned}&F(\boldsymbol{w}(t+1)) - F(\boldsymbol{w}(t))\\ \leq&-\gamma\left\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t) + \bar{\boldsymbol{e}}(t)\right\rangle + \frac{\gamma^2 S}{2}\|\boldsymbol{\Upsilon}(t) + \bar{\boldsymbol{e}}(t)\|_2^2\\ =&-\gamma\left\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t)\right\rangle - \gamma\left\langle\nabla F(\boldsymbol{w}(t)), \bar{\boldsymbol{e}}(t)\right\rangle\\ &+\frac{\gamma^2 S}{2}\|\boldsymbol{\Upsilon}(t)\|_2^2 + \frac{\gamma^2 S}{2}\|\bar{\boldsymbol{e}}(t)\|_2^2 + \gamma^2 S\left\langle\boldsymbol{\Upsilon}(t), \bar{\boldsymbol{e}}(t)\right\rangle\\ \overset{(a)}{\leq}&-\gamma\left\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t)\right\rangle + \frac{\gamma}{2}\|\nabla F(\boldsymbol{w}(t))\|_2^2 + \frac{\gamma}{2}\|\bar{\boldsymbol{e}}(t)\|_2^2\\ &+\frac{\gamma^2 S}{2}\|\boldsymbol{\Upsilon}(t)\|_2^2 + \frac{\gamma^2 S}{2}\|\bar{\boldsymbol{e}}(t)\|_2^2 + \gamma^2 S\left\langle\boldsymbol{\Upsilon}(t), \bar{\boldsymbol{e}}(t)\right\rangle\\ \overset{(b)}{\leq}&-\gamma\left\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t)\right\rangle + \frac{\gamma}{2}\|\nabla F(\boldsymbol{w}(t))\|_2^2\\ &+\left(\frac{\gamma}{2} + \gamma^2 S\right)\|\bar{\boldsymbol{e}}(t)\|_2^2 + \gamma^2 S\|\boldsymbol{\Upsilon}(t)\|_2^2,\end{aligned}$$

where (a) is due to $-\boldsymbol{a}^T\boldsymbol{b} \leq \frac{\|\boldsymbol{a}\|_2^2}{2} + \frac{\|\boldsymbol{b}\|_2^2}{2}$ and (b) is due to $\boldsymbol{a}^T\boldsymbol{b} \leq \frac{\|\boldsymbol{a}\|_2^2}{2} + \frac{\|\boldsymbol{b}\|_2^2}{2}$. By taking an expectation at both sides, we have

$$\begin{aligned}&\mathbb{E}\left[F(\boldsymbol{w}(t+1)) - F(\boldsymbol{w}(t))\right]\\ \leq& -\gamma\mathbb{E}[\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t)\rangle] + \frac{\gamma}{2}\|\nabla F(\boldsymbol{w}(t))\|_2^2\\ &+\left(\frac{\gamma}{2}+\gamma^2 S\right)\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2] + \gamma^2 S\mathbb{E}[\|\boldsymbol{\Upsilon}(t)\|_2^2].\end{aligned} \tag{26}$$

Note that $\boldsymbol{\Upsilon}(t) = \frac{1}{K}\sum_{k=1}^K \boldsymbol{\Upsilon}_k(t)$, we have

$$\begin{aligned}&\mathbb{E}\left[\left\langle\nabla F(\boldsymbol{w}(t)), \boldsymbol{\Upsilon}(t)\right\rangle\right] = \mathbb{E}\left[\left\langle\nabla F(\boldsymbol{w}(t)), \frac{1}{K}\sum_{k=1}^K \boldsymbol{\Upsilon}_k(t)\right\rangle\right]\\ =&\left[\left\langle\nabla F(\boldsymbol{w}(t)), \frac{1}{K}\sum_{k=1}^K \nabla F_k(\boldsymbol{w}(t))\right\rangle\right] = \|\nabla F(\boldsymbol{w}(t))\|_2^2,\end{aligned} \tag{27}$$

and

$$\begin{aligned}&\mathbb{E}\left[\|\boldsymbol{\Upsilon}(t)\|_2^2\right] = \mathbb{E}\left[\left\|\frac{1}{K}\sum_{k=1}^K \boldsymbol{\Upsilon}_k(t)\right\|_2^2\right]\\ \overset{(a)}{=}&\mathrm{Var}\left(\frac{1}{K}\sum_{k=1}^K \boldsymbol{\Upsilon}_k(t)\right) + \left\|\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^K \boldsymbol{\Upsilon}_k(t)\right]\right\|_2^2\\ \overset{(b)}{=}&\frac{1}{K^2}\sum_{k=1}^K \mathrm{Var}\left(\boldsymbol{\Upsilon}_k(t)\right) + \left\|\frac{1}{K}\sum_{k=1}^K \nabla F_k(\boldsymbol{w}(t))\right\|_2^2\\ \leq&\frac{1}{K}\xi^2 + \|\nabla F(\boldsymbol{w}(t))\|_2^2,\end{aligned} \tag{28}$$

where (a) holds because $\mathbb{E}[\|\boldsymbol{x}\|^2] = \mathrm{Var}[\boldsymbol{x}] + \|\mathbb{E}[\boldsymbol{x}]\|^2$, (b) follows from $\mathrm{Var}(\sum_{j=1}^n \boldsymbol{x}_j) = \sum_{j=1}^n \mathrm{Var}(\boldsymbol{x}_j)$ if $\{\boldsymbol{x}_j\}$ is independent. Hence, we have

$$\begin{aligned}&\mathbb{E}[F(\boldsymbol{w}(t+1)) - F(\boldsymbol{w}(t))\\ \leq& -\gamma\|\nabla F(\boldsymbol{w}(t))\|_2^2 + \frac{\gamma}{2}\|\nabla F(\boldsymbol{w}(t))\|_2^2\\ &+\gamma^2 S\left(\frac{1}{K}\xi^2 + \|\nabla F(\boldsymbol{w}(t))\|_2^2\right) + \left(\frac{\gamma}{2}+\gamma^2 S\right)\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2].\end{aligned} \tag{29}$$

By summing up above inequality for all T communication rounds, we have

$$\begin{aligned}&\mathbb{E}[F(\boldsymbol{w}(t)) - F(\mathbf{w}(0))]\\ \leq&\left(\frac{2\gamma^2 S-\gamma}{2}\right)\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{w}(t))\|_2^2\\ &+\frac{S\gamma^2\xi^2 T}{K} + \left(\frac{\gamma}{2}+\gamma^2 S\right)\sum_{t=0}^{T-1}\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2].\end{aligned}$$

With Assumption 1 and $\gamma < \frac{1}{2S}$, we have

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla F(\boldsymbol{w}(t))\|_2^2 \leq& \frac{2(F(\mathbf{w}(0) - F(\mathbf{w}^*)))}{\gamma(1-2\gamma S)T}\\ &+\frac{2S\gamma\xi^2}{K(1-2\gamma S)} + \frac{1+2\gamma S}{1-2\gamma S}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\bar{\boldsymbol{e}}(t)\|_2^2].\end{aligned}$$

REFERENCES

[1] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, 2022.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[3] Y. Shi, K. Yang, Z. Yang, and Y. Zhou, *Mobile Edge Artificial Intelligence: Opportunities and Challenges*. Elsevier, 2021.

[4] Y. Yang, Z. Zhang, and Q. Yang, "Communication-efficient federated learning with binary neural networks," *IEEE J. Sel. Area. Commun.*, vol. 39, no. 12, pp. 3836–3850, 2021.

[5] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Robust and communication-efficient federated learning from Non-i.i.d. data," *IEEE Trans. Neural Network. learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, 2020.

[6] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-Armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, 2020.

[7] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, 2020.

[8] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust federated learning with noisy communication," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3452–3464, 2020.

[9] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020.

[10] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, 2020.

[11] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2020.

[12] M. M. Wadu, S. Samarakoon, and M. Bennis, "Joint client scheduling and resource allocation under channel uncertainty in federated learning," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5962–5974, 2021.

[13] X. Cao, G. Zhu, J. Xu, and S. Cui, "Transmission power control for over-the-air federated averaging at network edge," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1571–1586, May 2022.

[14] X. Cao, G. Zhu, J. Xu, Z. Wang, and S. Cui, "Optimized power control design for over-the-air federated edge learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 342–358, Jan. 2022.

[15] 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, 2020.

[16] 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, 2020.

[17] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, 2021.

[18] Y. Yang, Y. Zhou, Y. Wu, and Y. Shi, "Differentially private federated learning via reconfigurable intelligent surface," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19 728 – 19 743, Oct. 2022.

[19] S. Samarakoon, J. Park, and M. Bennis, "Robust reconfigurable intelligent surfaces via invariant risk and causal representations," in *Proc. IEEE SPAWC*, Lucca, Italy, Sept. 2021, pp. 301–305.

[20] C. B. Issaid, S. Samarakoon, M. Bennis, and H. V. Poor, "Federated distributionally robust optimization for phase configuration of RISs," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[21] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, 2019.

[22] C. Huang, S. Hu, G. C. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. D. Renzo, and M. Debbah, "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 118–125, May 2020.

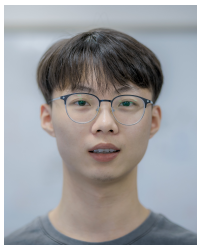
[23] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Area. Commun.*, vol. 38, no. 8, pp. 1839–1850, 2020.

[24] Z. Wang, J. Qiu, Y. Zhou, Y. Shi, L. Fu, W. Chen, and K. Letaief, "Federated learning via intelligent reflecting surface," *IEEE Trans. Wireless Commun*, 2021.

[25] H. Liu, X. Yuan, and Y. Zhang, "Reconfigurable intelligent surface enabled federated learning: A unified communication-learning design approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7595–7609, 2021.
[26] K. Yang, Y. Shi, Y. Zhou, Z. Yang, L. Fu, and W. Chen, "Federated machine learning for intelligent IoT via reconfigurable intelligent surface," *IEEE Network*, vol. 34, no. 5, pp. 16–22, Sept. 2020.
[27] D. Zhou, M. Sheng, Y. Wang, J. Li, and Z. Han, "Machine learning-based resource allocation in satellite networks supporting internet of remote things," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6606–6621, 2021.
[28] Y. Liu, X. Wang, J. Mei, G. Boudreau, H. Abou-Zeid, and A. Sediq, "Situation-aware resource allocation for multi-dimensional intelligent multiple access: A proactive deep learning framework," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 116–130, 2021.
[29] Z. Wang, J. Zong, Y. Zhou, Y. Shi, and V. W. S. Wong, "Decentralized multi-agent power control in wireless networks with frequency reuse," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1666–1681, Mar. 2022.
[30] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2268–2281, 2020.
[31] J. Tian, Q. Liu, H. Zhang, and D. Wu, "Multiagent deep-reinforcement-learning-based resource allocation for heterogeneous QoS guarantees for vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1683–1695, 2022.
[32] W. Lim, J. Ng, Z. Xiong, D. Niyato, C. Miao, and D. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, p. 3640–3653, 2021.
[33] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.
[34] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.
[35] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, "A GNN-based supervised learning framework for resource allocation in wireless IoT networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1712–1724, 2022.
[36] Z. Wang, Y. Zhou, Y. Shi, and W. Zhuang, "Interference management for over-the-air federated learning in multi-cell wireless networks," *IEEE J. Sel. Area. Commun.*, vol. 40, no. 8, pp. 2361–2377, 2022.
[37] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, 2021.
[38] X. Cao, G. Zhu, J. Xu, and K. Huang, "Optimized power control for over-the-air computation in fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7498 – 7513, Aug. 2020.
[39] Z. Wang, L. Liu, and S. Cui, "Channel estimation for intelligent reflecting surface assisted multiuser communications: Framework, algorithms, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6607–6620, 2020.
[40] Z. Mao, M. Peng, and X. Liu, "Channel estimation for reconfigurable intelligent surface assisted wireless communication systems in mobility scenarios," *China Commun.*, vol. 18, no. 3, pp. 29–38, 2021.
[41] G. Zhou, C. Pan, H. Ren, P. Popovski, and A. L. Swindlehurst, "Channel estimation for RIS-aided multiuser millimeter-wave systems," *IEEE Trans. Signal Process.*, vol. 70, pp. 1478–1492, 2022.
[42] S. Ma, W. Shen, X. Gao, and J. An, "Robust channel estimation for RIS-aided millimeter-wave system with RIS blockage," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5621–5626, 2022.
[43] T. Sery, N. Shlezinger, K. Cohen, and Y. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, 2021.
[44] Y. Zou, Z. Wang, X. Chen, H. Zhou, and Y. Zhou, "Knowledge-guided learning for transceiver design in over-the-air federated learning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2022.
[45] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Communi.*, vol. 54, no. 7, pp. 1310–1322, 2006.
[46] T. Jiang, H. V. Cheng, and W. Yu, "Learning to reflect and to beamform for intelligent reflecting surface with implicit channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, Jul. 2021.
[47] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. Int.l Conf. Learn. Represent. (ICLR)*, New Orleans, LA, May 2019.

Zixin Wang (Student Member, IEEE) received the B.S. degree from Wuhan University of Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He is currently a visiting student at University of Oulu. His research interests include edge intelligence and semantic communication.

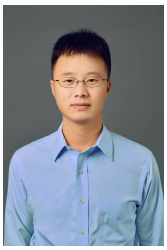
Yong Zhou (Senior Member, IEEE) received the B.Sc. and M.Eng. degrees from Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2015. From Nov. 2015 to Jan. 2018, he worked as a postdoctoral research fellow in the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. He is currently an Assistant Professor in the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He was the track co-chair of IEEE VTC 2020 Fall and the general co-chair of IEEE ICC 2022 workshop on edge artificial intelligence for 6G. His research interests include 6G communications, edge intelligence, and Internet of Things.



Yinan Zou (Student Member, IEEE) received the B.E. degree in electronic information engineering from Chongqing University, Chongqing, China, in 2020. He is currently pursuing the master's degree with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China.

Qiaochu An (Student Member, IEEE) received the B.S. degree from the School of Information Engineering, Zhengzhou University, Zhengzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests include over-the-air computation, intelligent reflecting surface, and federated learning.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2023.3239400



Yuanming Shi (Senior Member) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2011. He received the Ph.D. degree in electronic and computer engineering from The Hong Kong University of Science and Technology (HKUST), in 2015. Since September 2015, he has been with the School of Information Science and Technology in ShanghaiTech University, where he is currently a tenured Associate Professor. He visited University of California, Berkeley, CA, USA, from October 2016 to February 2017. His research areas include optimization, machine learning, wireless communications, and their applications to 6G, IoT, and edge AI. He was a recipient of the 2016 IEEE Marconi Prize Paper Award in Wireless Communications, the 2016 Young Author Best Paper Award by the IEEE Signal Processing Society, and the 2021 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He is also an editor of IEEE Transactions on Wireless Communications, IEEE Journal on Selected Areas in Communications, and Journal of Communications and Information Networks.

Mehdi Bennis (Fellow, IEEE) is a full (tenured) Professor at the Centre for Wireless Communications, University of Oulu, Finland and head of the intelligent connectivity and networks/systems group (ICON). His main research interests are in radio resource management, game theory and distributed AI in 5G/6G networks. He has published more than 200 research papers in international conferences, journals and book chapters. He has been the recipient of several prestigious awards including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best paper Award for the Journal of Wireless Communications and Networks, the all-University of Oulu award for research, the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award and the 2020 Clarviate Highly Cited Researcher by the Web of Science. Dr Bennis is an editor of IEEE TCOM and Specialty Chief Editor for Data Science for Communications in the Frontiers in Communications and Networks journal. Dr Bennis is an IEEE Fellow.