

# GraphRx: Graph-Based Collaborative Learning among Multiple Cells for Uplink Neural Receivers

Tianxin Wang\*, Xudong Wang<sup>†</sup>, Geoffrey Ye Li<sup>‡</sup>

\*Shanghai Jiao Tong University, Shanghai, China

<sup>†</sup>Hong Kong University of Science and Technology Guangzhou, Guangzhou, China

<sup>‡</sup>Imperial College London, London, United Kingdom

**Abstract**—A pre-trained neural receiver does not perform well in all channel environments, so online retraining is necessary. To acquire channel knowledge efficiently, collaborative learning among multiple neural receivers is indispensable. To this end, a graph-based collaborative learning scheme called GraphRx is developed to retrain uplink neural receivers collaboratively among base stations (BSs). First, considering a collaboration graph among BSs, GraphRx is formulated as a personalized federated learning problem, wherein the graph weights and neural receiver models are learned together so that generalization and personalization are jointly optimized. Second, the problem is solved through an alternating approach under the federated learning paradigm. Particularly, an approximate generalization bound is derived to enable graph optimization at the server without accessing local data on BSs. To reduce overhead of training pilots, data augmentation is employed. GraphRx is evaluated via extensive simulation. Key parameters of GraphRx are first found through ablation study. Next, the effectiveness of the approximation in the generation bound is validated. Comparisons with the state-of-the-art schemes are finally conducted. Results show that, given the same coded bit error rate, GraphRx achieves a SNR gain of 0.4 ~ 0.9 dB and 0.5 ~ 2.1 dB for the cases without and with inter-cell interference, respectively.

## I. INTRODUCTION

The sixth-generation (6G) communication networks are expected to be AI-empowered. Particularly, the neural network based wireless receiver, named *neural receiver*, is one of the most promising solutions for 6G receiver design [1]. As shown in Fig. 1, in a neural receiver, the channel estimator and detector are replaced by a neural network that takes the frequency-domain receive signals as inputs and outputs the soft-detection bits. By implicitly and accurately capturing real-world channels [1], neural receivers can significantly outperform the traditional ones.

In this paper, uplink neural receivers located at base stations (BS) are considered to detect the uplink orthogonal frequency division multiplexing (OFDM) signals from user terminals. As a common practice, a neural receiver is offline pretrained and then deployed online [2]. However, the pre-trained model may not generalize well to online channel environments, as there exists discrepancy in data distributions between offline and online environments [2], [3]. To this end, online knowledge acquisition for neural receivers becomes a necessity to allevi-

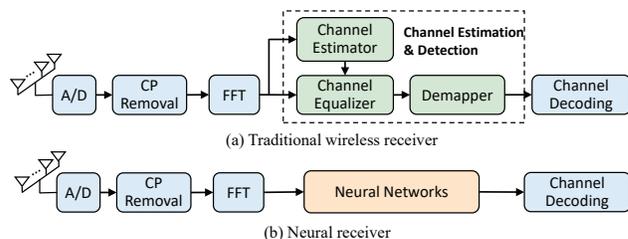


Fig. 1. Uplink neural receiver.

ate performance degradation in various channel environments. In other words, neural receivers need to be retrained online.

While online retraining of uplink neural receivers has been studied in [2]–[5], collaborative online retraining among multiple BSs has gained increasing attention recently [6]. In a collaborative learning framework, each BS collaborates with other BSs to learn its neural receiver under the coordination of a central server, such that the knowledge accumulated in the entire multi-cell network can be exploited. Multi-BS collaborative learning holds two key advantages as compared to single-BS local learning. First, it overcomes the problem of limited observation of channel instances in local learning. Through collaborative learning, the neural receiver in a cell enriches its online knowledge by learning data distributions from other cells, so the model generalization ability is enhanced. Second, collaborative learning helps neural receivers to improve the capability of handling inter-cell interference. As such interference encodes other cells’ channel information, a neural receiver with multi-cell channel knowledge can better grasp the statistical features of interfered signals.

To fulfill online collaborative learning for uplink neural receivers among multiple cells, it is necessary to design an appropriate collaboration mechanism. A straightforward one is to consider centralized learning where multi-cell data are gathered in one server. However, this strategy is not always desirable for two main reasons. First, there exists a privacy concern [7], as physical layer data at a BS are not supposed to be exposed to an entity in the cloud, unless a cloud radio access network (RAN) is considered. Second, centralized learning demands a costly server with abundant storage and computing resources. Therefore, it is more desirable to develop a distributed collaborative learning scheme that keeps the data locally. A common framework that can be adopted is federated learning (FL) [6], [8], where a global model is learned with

Corresponding author: Xudong Wang (Email: wxudong@ieee.org.) The research was sponsored by the National Natural Science Foundation of China (NSFC) Grant No. 62371292.

locally-held data. However, due to data heterogeneity across cells, the unified global model may perform even worse than the local models, which indicates a lack of personalization.

To achieve proper tradeoff between generalization and personalization, a graph-based collaborative learning scheme, called **GraphRx**, is developed for uplink neural receivers in this paper. First, GraphRx is formulated as a personalized federated learning (PFL) problem, based on a weighted collaboration graph where a node on the graph denotes a personalized model at a BS while a weighted edge represents the collaboration intensity between a pair of models. Moreover, the graph weights and neural receiver models are jointly optimized so that generalization and personalization are optimized together. Second, to solve the problem, a two-step alternating method is applied: 1) update the personalized model at each BS, given a collaboration graph and an aggregated model sent from the server; 2) determine the collaboration graph at the server, given the updated personalized models from all BSs. The first step is simple, but there exists a challenge in the second step, as the server has no access to local datasets on BSs, which prevents it from conducting graph optimization. Thus, the graph optimization problem in the second step is transformed so that it does not rely on local data. The key idea is to derive a generalization bound of the collaborative learning problem, approximate it empirically, and then replace the objective function of collaborative learning by the approximate bound. In this way, the collaboration graph can be determined at the server without using local data of BSs.

As compared to the existing graph-based PFL algorithm [9], GraphRx is distinct in three aspects: 1) collaboration intensity is captured with finer granularity rather than just model cosine similarity; 2) no regularization with graph weights is necessary; 3) the collaboration graph is optimized by following the generalization bound of multi-source domain adaptation.

To support online training, pilot-free methods in [3], [5] can be used where labels are directly retrieved from communication data, but their performance can be easily impacted by erroneous labels. Thus, pilot-based training sequences (i.e., *training pilots*) are adopted to retrain the neural receiver in this paper. To reduce overhead of training pilots, data augmentation is needed [3]. Specifically, two techniques, rotation and noise injection, are employed at each BS to enrich the quantity and diversity of online training data.

The main contributions of this paper are summarized as follows:

- A collaborative learning scheme, called GraphRx, is proposed based on a collaboration graph among BSs to retrain uplink neural receivers in multiple cells together. It acquires channel knowledge in multiple cells collaboratively and handles multi-cell interference effectively.
- GraphRx is formulated as a PFL problem, wherein the personalized models of neural receivers and the collaboration graph are optimized jointly to achieve tradeoff between generalization and personalization.
- The solution to the PFL problem is derived and mapped to the federated learning paradigm as an iterative algorithm

alternating between the server and BSs. Particularly, an approximate generalization bound is derived so that the server can determine the collaboration graph without relying on local data of BSs.

- Extensive simulation is carried out to validate key parameters and mechanisms of GraphRx. Performance results show that GraphRx significantly outperforms the state-of-the-art schemes.

The rest of this paper is organized as follows. Related work is presented in Section II, while the system model is provided in Section III. The problem formulation and analysis are given in Section IV. After the issue of collaboration graph optimization is resolved in Section V, the entire scheme of GraphRx is elaborated in Section VI. Performance evaluation is carried out in Section VII, and the paper is concluded in Section VIII.

## II. RELATED WORK

Compared with traditional model-based receivers, neural receivers can: 1) reduce the overhead incurred by reference signals to achieve a higher spectral efficiency [10]; 2) adapt to real-world wireless channels without knowing underlying channel models [1], [11]; 3) learn to compensate for hardware-induced non-linear distortions [12], [13]. There exist various architectures for neural receivers. As an early study in this field, the OFDM neural receiver is designed with a fully-connected neural network in [13]. Data-driven end-to-end OFDM systems are further developed in [14], [15] to jointly learn the transmitter and receiver. Furthermore, recurrent networks [16] and convolution networks [1], [17] are also employed in neural receivers. Since the model architecture in [1] matches the physical layer setup of this paper, it is adopted as a basic building block of neural receivers.

Online adaptation/retraining techniques for neural receivers are studied in [2]–[5]. However, limited online channel knowledge at a single receiver restrains model generalization ability, which calls for collaborative learning among multiple receivers.

As a collaborative learning framework, federated learning (FL) [8], [18] enables joint training of models at multiple clients with privacy-preserving properties. To address the challenge of data heterogeneity in FL, personalized FL (PFL) [19] has been developed to enable each client to train a personalized model during collaboration, which generally falls into two categories [20]: coarse-grained PFL and fine-grained PFL. For coarse-grained PFL, the algorithms cannot determine the collaboration structure among clients (i.e., which clients to collaborate on what level). Ditto [21] uses the globally-shared model as regularization to develop personalized models, while FedRep and its variants [22], [23] share one feature extractor globally and personalize the last few layers of models. For fine-grained PFL [9], [24], there exists a collaboration structure that can be flexibly adjusted to diverse types of heterogeneous data distributions. Similar to our scheme GraphRx, the state-of-the-art fine-grained method pFedGraph [9] also uses a graph to represent pair-wise collaboration among clients.

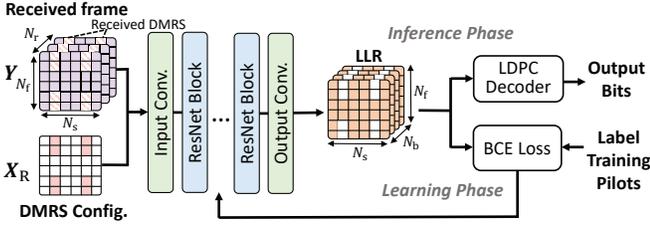


Fig. 2. Fully convolutional network architecture of neural receiver.

However, our scheme features a different problem formulation and also a rigorous solution derived from the generalization bound. Additionally, GraphRx is the first neural receiver designed based on a fine-grained PFL.

### III. SYSTEM MODEL

#### A. SIMO-OFDM Communications

The uplink SIMO OFDM communication is considered with  $N_r$  receive antennas at the BS and one transmit antenna at the user. One OFDM frame spans a transmission time interval (TTI) and includes  $N_s$  OFDM symbols and  $N_f$  subcarriers. Each resource element (RE) has one symbol time and one subcarrier. Let  $\mathbf{Y}, \mathbf{H}, \mathbf{N}, \mathbf{G} \in \mathbb{C}^{N_r \times N_s \times N_f}$  be the received signals, the channel coefficients, the additive white Gaussian noise, and the inter-cell interference of one OFDM frame, respectively. The transmit signal matrix is  $\mathbf{X} \in \mathbb{C}^{N_s \times N_f}$ . Let  $k$  be the index of OFDM symbol and  $l$  be the index of subcarriers. We define the following vectors  $\mathbf{y}_{kl} = \mathbf{Y}[:, k, l]$ ,  $\mathbf{h}_{kl} = \mathbf{H}[:, k, l]$ ,  $\mathbf{n}_{kl} = \mathbf{N}[:, k, l]$ ,  $\mathbf{g}_{kl} = \mathbf{G}[:, k, l]$  and they all have the dimension of  $N_r$ . The frequency-domain received signal on each RE during one TTI is

$$\mathbf{y}_{kl} = \mathbf{h}_{kl}x_{kl} + \mathbf{g}_{kl} + \mathbf{n}_{kl}, \quad (1)$$

where  $x_{kl} \in \mathbf{X}, \forall k = 1, \dots, N_s, j = 1, \dots, N_f$ .

#### B. Design of Neural Receiver

As shown in Fig. 2, for the model architecture of neural receiver, a fully convolutional neural network (CNN) consisting of multiple preactivation ResNet blocks [1] is employed. The input of neural receiver has two parts: received OFDM frame  $\mathbf{Y} \in \mathbb{C}^{N_r \times N_s \times N_f}$  and transmitted demodulation reference symbols (DMRS) configuration matrix  $\mathbf{X}_R \in \mathbb{C}^{N_s \times N_f}$ . Aligned with standards [25], DMRS occupies selected subcarriers in one or two symbol times within each TTI. To match the dimension of  $N_s \times N_f$ , zeros are inserted in non-DMRS positions within the frame to form  $\mathbf{X}_R$ . By stacking  $\mathbf{Y}$  and  $\mathbf{X}_R$ , we have  $\mathbf{T} \in \mathbb{R}^{(2N_r+2) \times N_s \times N_f}$  by treating their real and imaginary parts as two separate channels. The output  $h_{\theta}(\mathbf{T}) \in \mathbb{R}^{N_s \times N_f \times N_b}$  is the log-likelihood ratios (LLR) with  $N_{RS}$  symbols in DMRS positions ignored, where  $N_{RS}$  is the number of DMRS symbols within one TTI and  $N_b$  is the number of bits per symbol.

For model inference,  $\mathbf{Y}$  contains received uplink data except REs in DMRS positions. The model output (i.e., LLR of data bits) is directly fed into an LDPC channel decoder to obtain detected bits. For model training,  $\mathbf{Y}$  contains *received*

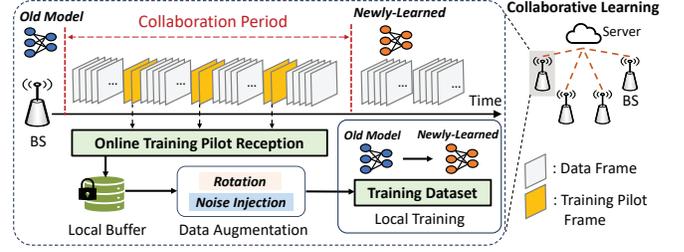


Fig. 3. Working process of each BS in collaborative learning.

*training pilots* that are transmitted in place of uplink data. Since bit detection is a binary classification problem for each bit, the model output  $h_{\theta}(\mathbf{T})$  is compared against label  $\mathbf{L}$  to compute the binary cross-entropy (BCE) loss after Sigmoid activation [1], where  $\mathbf{L}$  is the preknown training pilot bits of length  $(N_s N_f - N_{RS})N_b$ . In this paper, transmission and reception of training pilots specially serve for model retraining, while system-configured DMRS are kept for a neural receiver to implicitly perform channel estimation by using  $\mathbf{X}_R$  and received DMRS symbols in the corresponding positions of  $\mathbf{Y}$  [1].

#### C. Elements of Supervised Learning Problem

There are  $M$  base stations and one central server in a multi-cell network shown in Fig. 3. The set of BS indices is denoted by  $\mathcal{M} = \{1, \dots, M\}$ . An OFDM frame carrying training pilots is named *training pilot frame* while a normal frame carrying communication data is named *data frame*. BS  $m$  builds up an online dataset  $\mathcal{Z}_m$  locally via periodic reception of training pilot frames [1]. Local dataset  $\mathcal{Z}_m = \{(\mathbf{T}_m^{(i)}, \mathbf{L}_m^{(i)})\}_{i=1}^{N_m}$  has  $N_m$  data instances, where  $\mathbf{T}_m^{(i)}$  and  $\mathbf{L}_m^{(i)}$  are the input and label of the  $i$ -th data instance. Let  $\mathbf{p} = [p_1, \dots, p_M]$  be the data quantity vector, i.e.,  $p_m = \frac{N_m}{N}$ , where  $N = \sum_{m=1}^M N_m$  denotes the total number of data instances. Empirical risk function  $\hat{F}(\cdot)$  over local dataset  $\mathcal{Z}_m$  is defined as

$$\hat{F}(\theta; \mathcal{Z}_m) = \frac{1}{N_m} \sum_{i=1}^{N_m} \ell \left( h_{\theta} \left( \mathbf{T}_m^{(i)} \right), \mathbf{L}_m^{(i)} \right), \quad (2)$$

where  $h_{\theta}$  is a mapping function parameterized by model parameters  $\theta$ , and  $\ell(\cdot)$  is the BCE loss stated previously. In this paper, each BS has a personalized neural receiver denoted by  $\theta_m, \forall m \in \mathcal{M}$ .

### IV. PROBLEM FORMULATION AND ANALYSIS

The overall working procedure of each BS is illustrated in Fig. 3. During online deployment, each BS collects a local dataset via periodic reception of training pilot frames within a fixed time interval named *collaboration period*. To avoid disrupting normal communications, training pilot frames are sparsely inserted among data frames, such as at a ratio of one training pilot frame for every 1000 data frames. This local dataset is further augmented to enrich its quantity and diversity, as elaborated in Section VI-B.

Based on the collected data, each BS participates in multi-cell collaborative learning coordinated by the cloud server,

such that it learns data distributions from other cells. It is emphasized that collaborative learning is a large-timescale task [26] running in the background, as sufficient online data must be collected for a neural receiver to learn statistical features of online channels effectively. In addition, collaborative learning will not disrupt online inference of the currently deployed neural receiver. Once a new neural receiver is learned via collaborative learning, it replaces the old one in operation. Local real-time adaptation of neural receiver [3], [5] may be incorporated in the framework to temporarily enhance performance, but it is not the focus of this paper. Next, the graph-based collaborative learning problem is formulated.

### A. Collaboration-Graph-Based Learning

To reveal the pair-wise collaboration formed among BSs, a directed collaboration graph weights  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  is defined: node set  $\mathcal{V} = \{\theta_m\}_{m \in \mathcal{M}}$  is the set of local models, and edge set  $\mathcal{E} = \{(j, m) | w_{mj} \geq 0, \forall m, j \in \mathcal{M}\}$ , where  $w_{mj}$  is the edge weight on the directed edge from node  $j$  to node  $m$ . This edge weight also indicates on what level node  $m$  should learn from the data distribution of node  $j$ . There is no collaboration between two nodes if  $w_{mj} = 0$ . Let  $\mathbf{W}$  be the directed graph weight matrix with  $w_{mj}$  as its element. Let  $\mathbf{w}_m = [w_{m1}, \dots, w_{mM}]$  be the collaboration vector of BS  $m$ , and  $\mathcal{N}_m = \{j | j \neq m, w_{mj} > 0, j \in \mathcal{M}\}$  be the neighbor set of BS  $m$ .

Collaboration graph weights  $\mathbf{W}$  are unknown and thus must be optimized along with personalized model parameters  $\{\theta_j\}_{j \in \mathcal{M}}$ . The existing graph-based PFL formulation [9] embeds the graph weights into both the aggregated model term and the graph regularization term, which has two problems: 1) the second term is redundant when the first term already exists; 2) the second term based on cosine similarity only measures the angular difference between model vectors but ignores their magnitude difference.

To this end, a new formulation for graph-based learning is designed to explicitly capture the data-level collaboration from the perspective of each BS: BS  $m$  trains its personalized model  $\theta_m$  by learning from local data  $\mathcal{Z}_m$  and diverse data from its neighbors  $\mathcal{Z}_j, j \in \mathcal{N}_m$ , such that the learned personalized model enjoys better generalization ability; meanwhile, collaboration weight  $w_{mj}$  controls how much diverse data distributions contribute to the model learning such that collaboration will not cause local performance degradation.

Based on the above design rational, the graph-based collaborative learning of personalized model  $\theta_m, \forall m \in \mathcal{M}$  is formulated as Problem 1 (P1):

$$\theta_m, \mathbf{w}_m = \arg \min_{\theta, \mathbf{w}_m} \sum_{j \in \{m\} \cup \mathcal{N}_m} w_{mj} \hat{F}(\theta; \mathcal{Z}_j) \quad (3)$$

$$s.t. \quad \sum_{j=1}^M w_{mj} = 1, w_{mj} \geq 0, \forall j \in \mathcal{M}, \quad (4)$$

where the sum of non-negative weights in a collaboration vector is normalized to 1 in (4). The collaboration graph can flexibly balance the contribution of local knowledge

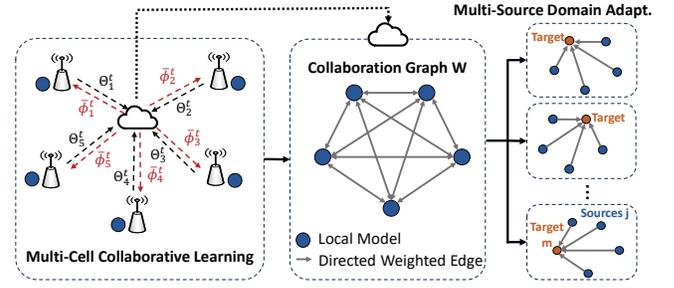


Fig. 4. GraphRx: graph-based collaborative learning scheme.

and collaborative knowledge to capture the personalization-generalization tradeoff. If the collaborative-knowledge term denoted by  $\sum_{j \in \mathcal{N}_m} w_{mj} \hat{F}(\theta; \mathcal{Z}_j)$  is excluded from the objective (3), P1 degenerates to a local learning problem without inter-BS collaboration. On the other hand, if the graph weights  $\mathbf{W}$  is set to be fully-connected with  $\mathbf{w}_m = p_m \mathbf{1}$  and there exists only one universal model, P1 reduces to a conventional FL problem [8].

### B. Problem Analysis

Assuming a centralized setting where all the data and models are collected to the server, the server can directly solve P1 for each BS in parallel. However, the FL setting in our problem imposes two critical constraints [8]: 1) only models are communicated between the server and BSs; 2) there is no peer-to-peer communication between BSs. These constraints render solving P1 for each BS much more challenging. Following the principle of optimization decomposition, P1 is decomposed into two alternative steps: 1) at each BS  $m$ , optimizing local model  $\theta_m$  with collaboration graph weights  $\mathbf{W}$  fixed; 2) at the server, optimizing collaboration graph weights  $\mathbf{W}$  given all the collected local models  $\{\theta_j\}_{j \in \mathcal{M}}$ .

For the first step, to fulfill collaboration while preserving data locality, local model update is combined with weighted model aggregation. Note that model aggregation is performed based on  $\mathbf{W}$  at the server side. As shown in Fig. 4, each BS  $m$  first fetches an aggregated model denoted as  $\bar{\phi}_m$  from the server as local model initialization. Each BS then runs local gradient descent to update its local model. The details will be elaborated in Section VI-A.

For the second step, however, it stills remains a critical challenge to determine the collaboration graph. The server can only access models  $\{\theta_m\}_{m \in \mathcal{M}}$  while it cannot access local data to evaluate empirical risks  $\hat{F}(\cdot)$ . Therefore, objective (3) cannot be optimized directly. In the following section, a new methodology derived from the generalization bound theory is designed.

## V. COLLABORATION GRAPH OPTIMIZATION

To optimize the collaboration graph, the theoretical generalization bound of  $\{\theta_m\}_{m \in \mathcal{M}}$  is first derived. It is expressed by a function of collaboration graph weights  $\mathbf{W}$ . Next, the bound is approximated empirically given the current personalized models, and the approximate bound is used as an empirical objective for optimizing graph weights  $\mathbf{W}$ .

### A. Derivation of Generalization Bound

In the rest of the paper, let  $\hat{F}_m(h) = \hat{F}(h; \mathcal{Z}_m)$  for ease of notation. In preparation for theoretical derivation, let a hypothesis function, denoted by  $h \in \mathcal{H}$ , be a general mapping from the model's input to its output, where  $\mathcal{H}$  is the hypothesis space quantified by finite Vapnik-Chervonenkis (VC) dimension  $c$  [27]. Let  $F_m(h)$  be the expected risk of a hypothesis  $h$  over true data distribution  $\mathcal{D}_m$  at BS  $m$ , that is,  $F_m(h) = \mathbb{E}_{\mathbf{T} \sim \mathcal{D}_m}[\ell(h(\mathbf{T}), \mathbf{L})]$  where  $\mathbf{T}$  is sampled from  $\mathcal{D}_m$ .

The fundamental goal of collaborative learning is to improve the generalization ability of each hypothesis to unseen data instances, which is equivalent to minimizing the expected risk denoted by  $F_m(\cdot)$ . However, in practice, true data distribution  $\mathcal{D}_m$  cannot be known. Therefore, each node has to minimize the empirical risk (i.e., objective (3)) over limited data instances from a variety of data sources. Given any collaboration vector  $\mathbf{w}_m$  that satisfies constraint (4), the objective of each BS  $m$  in (3) is expressed as minimizing the empirical  $\mathbf{w}_m$ -weighted error  $\hat{F}_{\mathbf{w}_m}(h)$ :

$$\hat{h}_{\mathbf{w}_m} = \arg \min_{h \in \mathcal{H}} \hat{F}_{\mathbf{w}_m}(h), \quad (5)$$

where  $\hat{F}_{\mathbf{w}_m}(h) = \sum_{j=1}^M w_{mj} \hat{F}_j(h)$ . To evaluate how  $\hat{h}_{\mathbf{w}_m}$  generalizes over the true data distribution, the generalization bound of  $\hat{h}_{\mathbf{w}_m}$  is expressed as the upper bound of the expected risk, i.e.,  $F_m(\hat{h}_{\mathbf{w}_m})$  [27]. This generalization bound is derived in the following theorem.

**Theorem 1.**  $\mathcal{H}$  is a hypothesis space of VC dimension  $c$ . For each  $j \in \mathcal{M}$ ,  $N \cdot p_j$  data points are drawn from distribution  $\mathcal{D}_j$ . For any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ , there exists

$$F_m(\hat{h}_{\mathbf{w}_m}) \leq F_m(h_m^*) + 2 \sqrt{\sum_{j=1}^M \frac{w_{mj}^2}{p_j} \left( \frac{c \log(2N) - \log \delta}{2N} \right)} + \sum_{j=1}^M w_{mj} (d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_m, \mathcal{D}_j) + 2\lambda_{mj}),$$

where  $h_m^* = \arg \min_{h \in \mathcal{H}} F_m(h)$  is the optimal minimizer of expected risk, and  $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_m, \mathcal{D}_j)$  is the  $\mathcal{H}\Delta\mathcal{H}$ -divergence [27], and  $\lambda_{mj} = \min_{h \in \mathcal{H}} (F_m(h) + F_j(h))$ .

*Proof.* One can view the optimization problem in (5) as a multi-source domain adaptation (DA) problem [27]. As shown in Fig. 4, the data instances of BS  $m$  are sampled from target domain  $\mathcal{D}_m$  and the data instances of BS  $j \in \mathcal{N}_m$  are sampled from source domains  $\{\mathcal{D}_j | j \in \mathcal{N}_m\}$ . Collaboration vector  $\mathbf{w}_m$  is then the domain weights. The objective of multi-source DA is to minimize the expected risk in the target domain, which is equivalent to (5). Thus, the generalization bound of multi-source DA (Theorem 3 in [28]) can be applied. Detailed derivations are omitted due to space limit.  $\square$

For ease of notation, let  $d_{mj} = d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_m, \mathcal{D}_j) + 2\lambda_{mj}$ , and  $\mathbf{D}$  is defined as the distribution divergence matrix whose element at the  $m$ -th row and the  $j$ -th column is pair-wise

divergence  $d_{mj}$ . Inspired by multi-source DA, for each node on the collaboration graph, it can be considered in a target domain with its neighbor nodes in source domains. Thus, a graph-based learning problem can be decomposed into the parallel multi-source DA processes shown in Fig. 4. The generalization bound in Theorem 1 is derived for each BS  $m$ , and thus the total generalization bound is obtained via the summation of all the upper bounds:

$$\sum_{m=1}^M F_m(\hat{h}_{\mathbf{w}_m}) \leq \sum_{m=1}^M F_m(h_m^*) + 2Q(\mathbf{W}, \mathbf{p}, N, c) + \sum_{m,j \in \mathcal{M}} [\mathbf{W} \circ \mathbf{D}]_{mj}, \quad (6)$$

where  $Q(\mathbf{W}, \mathbf{p}, N, c) = \sum_{m=1}^M \sqrt{\sum_{j=1}^M \frac{w_{mj}^2}{p_j} \left( \frac{c \log(2N) - \log \delta}{2N} \right)}$ ,  $\sum_{m,j \in \mathcal{M}} [\mathbf{W} \circ \mathbf{D}]_{mj} = \sum_{m=1}^M \sum_{j=1}^M w_{mj} d_{mj}$ .

The total generalization bound shown in Eq. (6) contains three terms: 1) the first term,  $\sum_{m=1}^M F_m(h_m^*)$ , is deemed as a constant value irrelevant to the graph weights; 2) the second term,  $2Q(\mathbf{W}, \mathbf{p}, N, c)$ , is a *quantity-aware term* that depends on data quantity vector  $\mathbf{p}$ , graph weights  $\mathbf{W}$ , total number of data instances  $N$ , and VC dimension  $c$  of the hypothesis space; 3) the third term,  $\sum_{m,j \in \mathcal{M}} [\mathbf{W} \circ \mathbf{D}]_{mj}$ , is the *distribution-divergence-aware term* depending on collaboration weights  $\mathbf{W}$  and distribution divergence matrix  $\mathbf{D}$ .

When the theoretical generalization bound (right-hand-side of Eq. (6)) is minimized, the total expected risk of all the hypothesis functions has the lowest upper bound, leading to the best generalization ability. Therefore, the fundamental insight is to find a collaboration graph that minimizes the generalization bound in Eq. (6), and thus the obtained graph can achieve the best generalization ability for personalized models. Specifically, within the bound, the quantity-aware term and the distribution-divergence-aware term (both are dependent on  $\mathbf{W}$ ) need to be minimized. However, due to the difficulty of computing these two theoretical terms without data, they must be approximated such that an empirical objective of bound minimization can be formed at the server side.

### B. Empirical Objective for Optimizing Collaboration Graph

To find an empirical objective for optimizing the collaboration graph, three steps are taken. First, the quantity-aware term in Eq. (6) is approximated. Complexity indicator is defined as  $C = 2\sqrt{(c \log(2N) - \log \delta)/2N}$ . It is treated as a hyper-parameter embedded in the quantity-aware term.

Second, the distribution-divergence-aware term in Eq. (6) is approximated. Pair-wise divergence  $d_{mj}$  is approximated by the distance between the two models,  $\hat{d}(\theta_m, \theta_j)$ . Note that this approximation of distribution divergence is adaptively adjusted as the model parameters are iteratively updated. The positive correlations between the distribution divergence and the model difference are observed in [9], [29], and it is also proved in [18] that the statistical distance between two distributions directly causes model parameter difference. In Section VII-C,

this approximation is validated by comparing model difference with estimated distribution divergence.

Third, combining the above approximate terms, an empirical optimization objective regarding  $\mathbf{W}$  is formed as

$$\hat{B}(\mathbf{W}, \Theta) = C \sum_{m=1}^M \sqrt{\sum_{j=1}^M \frac{w_{mj}^2}{p_j}} + \sum_{m=1}^M \sum_{j=1}^M w_{mj} \hat{d}(\theta_m, \theta_j). \quad (7)$$

By minimizing (7), the optimal collaboration graph is obtained, leading to the lowest approximate generalization bound.

## VI. GRAPH-BASED COLLABORATIVE LEARNING

Two alternating steps stated in Section IV are elaborated. In addition, the data augmentation mechanism is adopted to enrich training data.

### A. Alternating Optimization

The designed collaborative learning process consists of the alternation of two steps: 1) at each BS  $m$ , optimizing the local model  $\theta_m$  with collaboration graph weights  $\mathbf{W}$  fixed; 2) at the server, optimizing collaboration graph weights  $\mathbf{W}$  given all the local models  $\{\theta_m\}_{m \in \mathcal{M}}$ . Let  $t$  denote the index of communication round.

1) *Optimizing local model at each BS:* The distributed model updating process follows a widely-used protocol in distributed learning. Each BS first fetches the aggregated model denoted by  $\bar{\phi}_m^t$  from the server. Each BS then uses it to initialize the local model as

$$\theta_m^{t,0} \leftarrow \bar{\phi}_m^t, \quad (8)$$

where  $\theta_m^{t,0}$  is the initial model. Suppose  $\theta_m^{t,i}$  is the model after the  $i$ -th mini-batch gradient descent, then it is updated by minimizing local empirical risk  $\hat{F}_m(\cdot)$  in the next step as

$$\theta_m^{t,i+1} \leftarrow \theta_m^{t,i} - \eta \nabla \hat{F}_m(\theta_m^{t,i}), \quad (9)$$

with  $\eta$  as the learning rate. After  $s$  steps starting from  $\theta_m^{t,0}$ , each BS sends the updated local model  $\theta_m^{t,s}$  to the server.

2) *Optimizing collaboration graph at the server:* As stated in Section V, the empirical objective (7) guided by the generalization bound is minimized:

$$\mathbf{W}^t = \arg \min_{\mathbf{W}} \hat{B}(\mathbf{W}, \Theta) \quad (10)$$

$$s.t. \quad \sum_{j=1}^M w_{mj} = 1, w_{mj} \geq 0, \forall j, m \in \mathcal{M}. \quad (11)$$

The above problem is a convex optimization, and thus its global optimum is solved by conventional solvers [30]. Once the current collaboration graph weights  $\mathbf{W}^t$  for  $\{\theta_m^{t,s}\}_{m \in \mathcal{M}}$  is obtained, the server computes the aggregated model for each BS  $m$ , which is expressed by

$$\bar{\phi}_m^{t+1} = \sum_{j \in \{m\} \cup \mathcal{N}_m} w_{mj}^t \theta_j^{t,s}, \forall m \in \mathcal{M}, \quad (12)$$

which will be sent to each BS  $m$  in the next round.

Based on the above two steps, such  $T$  communication rounds of training are performed until each BS  $m$  obtains a well-learned personalized neural receiver  $\theta_m^T$  with multi-cell collaborative knowledge and the learned collaboration graph is  $\mathbf{W}^T$ .

### B. Mechanism of Data Augmentation

To achieve high spectral efficiency, each BS must keep the overhead of training pilots low. Thus, the overall quantity of training pilots may not be sufficient to produce satisfying retraining results within a relatively short collaboration period. To this end, two data augmentation techniques, rotation and noise injection, are employed at each BS to enrich the quantity and diversity of online training data.

Specifically, we adopt the design in [3]: 1) a rotating angle  $\phi$  is uniformly sampled from  $[0, 2\pi)$ , and each symbol in received training pilot frame  $\mathbf{Y}$  is applied such a rotation to obtain  $e^{j\phi} \mathbf{Y}$ ; 2) an additive noise  $\mathbf{N}_a$  (with the same dimension as  $\mathbf{N}$ ) is injected to each symbol, where  $\mathbf{N}_a \sim \mathcal{CN}(0, \sigma_a^2)$ .  $\sigma_a^2$  is uniformly sampled from  $(0, \frac{\sigma^2}{2}]$  ( $\sigma^2$  is the noise variance of  $\mathbf{N}$ ). Thus, the augmented frame (denoted as  $\mathbf{Y}_a$ ) is obtained:  $\mathbf{Y}_a = e^{j\phi} \mathbf{Y} + \mathbf{N}_a$ . A new instance can be created by replacing  $\mathbf{Y}$  with  $\mathbf{Y}_a$  during training with the label unchanged, and  $\mathbf{Y}_a$  will not be stored. Thus, no additional storage cost is incurred. If the above augmentation is applied  $A - 1$  times for each original instance, the augmented training data quantity is increased by  $A$  times (i.e., augmentation time).

## VII. PERFORMANCE EVALUATION

In this section, the simulation set-up is first presented, followed by the results of ablation study, design verification, and comparisons with several baselines.

### A. Simulation Set-up

For uplink SIMO communications, we assume one antenna at the user terminal and two receive antennas at the BS. The number of BSs in multi-cell collaboration is set to 6, i.e.,  $M = 6$ . Heterogeneous channel environments are assumed for these six cells. Specifically, the tapped delay line (TDL) channel models with diverse power delay profiles (PDP) [31], different values of fast-fading parameters [3] (i.e., root-mean-squared (RMS) delay spread  $\sigma_{ds}$  and terminal velocity  $v$ ) are adopted. The training pilots are random binary sequences and DMRS is configured at the second and eleventh symbol times within each TTI [25]. The offline channel distribution is specified as the TDL-A profile in the RMS delay spread interval of 0-50 ns and the velocity interval of 0-5 m/s over the SNR range [-4, 15] dB. The neural receiver is pretrained over 64000 TTIs offline with the mini-batch size set to 32, and then deployed online as the initial model for GraphRx. To generate each channel instance, the exact values of  $t_{ds}$ ,  $v$ , and SNR are uniformly sampled from the given intervals.

For online multi-cell channel generation, there are two types of online heterogeneous channel distributions among six cells as shown in Table I, namely heterogeneous data distributions 1 and 2 (i.e., Hetero-1 and Hetero-2). In the first type, within

TABLE I  
HETEROGENEOUS MULTI-CELL ENVIRONMENTS

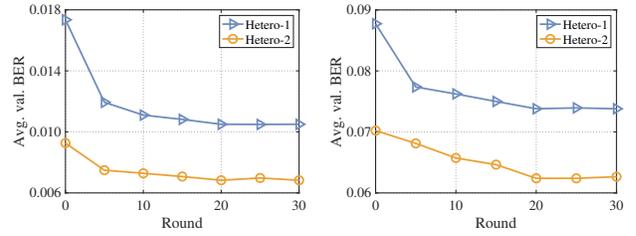
BS ID	Hetero-1			Hetero-2		
	PDP	$\sigma_{ds}(ns)$	$v(m/s)$	PDP	$\sigma_{ds}(ns)$	$v(m/s)$
1	B	500-600	0-5	B	0-50	0-5,15-20
2	B	450-550	0-5	C	450-500	0-5,15-20
3	C	200-300	15-20	B	0-50	0-5
4	C	150-250	15-20	C	150-200	0-5
5	E	400-600	15-20	E	0-50	0-5,15-20
6	E	450-550	15-20	E	200-400	0-5,15-20

each of these three pairs (BSs 1 and 2, BSs 3 and 4, BSs 5 and 6), two BSs exhibit similar channel distributions. Each BS holds the same training data quantity. In the second type, a more complicated heterogeneity is presented, and the training data quantity ratio of six BSs is 3:3:1:1:2:4. The total quantity of multi-cell training data is 6400 TTIs. The online SNR range is set to [0,10] dB, and the mini-batch size is 32.

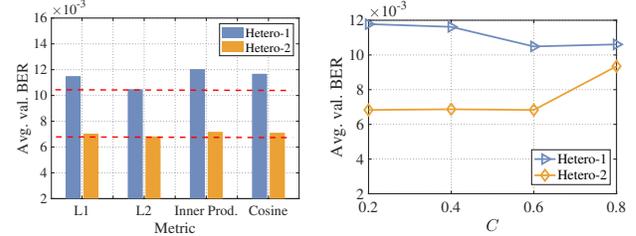
In addition, inter-cell interference is generated by letting another signal propagate through an interfering channel and reach the interfered BS [1]. For each uplink of BS  $m$ , the interfering PDP is randomly selected among all the other BSs' channel profiles, and RMS delay spread of interfering channel has a deviation of  $\pm 20$  ns from that of an uplink channel. The interference power is on average 5 dB lower than the noise power [32]. Note that the above interfering channel information is not known by the receiver.

The critical simulation parameters are as follows. In each cell, the system carrier frequency is 4 GHz with subcarrier frequency 15kHz and OFDM symbol duration  $71\mu s$ , and  $N_f = 72, N_s = 14, N_{RS} = 48$ . The test modulation scheme is 16-QAM and LDPC channel coding with the code rate as 658/1024 is adopted. Unless otherwise specified, the hyper-parameters of GraphRx are as follows: the number of epochs for local gradient descent is set to 2; initial collaboration graph weights  $\mathbf{W}^0 = \frac{1}{6}\mathbf{I}$ ; complexity indicator  $C = 0.6$ , data augmentation time  $A = 4$ , number of communication rounds  $T = 20$ , which are confirmed via ablation study. The  $\ell_2$ -norm is adopted as the model distance metric stated in Section V-B, i.e.,  $\hat{d}(\theta_m, \theta_j) = \|\theta_m - \theta_j\|_2$ . Each distance value is scaled into [0, 1]. For the training of neural receiver, the Adam optimizer with learning rate  $\eta = 0.001$  is adopted.

The main performance metric for evaluating receiver performance is coded bit-error rate (BER). Our method GraphRx is compared with the following baselines: 1) practical LMMSE receiver [1]; 2) the locally-trained neural receivers with no collaboration; 3) the FedAvg-trained globally-unified neural receiver [8]; 4) the neural receivers trained by the state-of-the-art PFL algorithms. Specifically, three representative PFL algorithms are selected: Ditto [21], FedRep [22], and pFedGraph [9]. Furthermore, the genie-aided LMMSE receiver with full and perfect channel coefficients is considered to achieve the upper-bound BER performance in the interference-free case. During training, a single-point BER (*validation BER*) is evaluated over 20% of training data. To test performance of



(a) Without interference (b) With inter-cell interference  
Fig. 5. Average validation BER at various communication rounds.



(a) Model distance metric (b) Complexity indicator value  
Fig. 6. Average validation BER with various distance metrics and  $C$ .

a trained neural receiver, the standard Monte Carlo simulation [25] is performed across the SNR range of [-4,15] dB based on Table I.

## B. Ablation Study

1) *Impact of communication round*: The number of communication rounds for GraphRx to converge is explored. During training, the validation BER is evaluated at 4 dB for each neural receiver. As shown in Fig. 5, in both interference-free and interference-included scenarios, GraphRx reaches convergence at the 20-th communication round, where the average validation BER of all BSs remains stable thereafter. The existence of training data quantity imbalance and inter-cell interference does not affect the convergence of GraphRx.

2) *Impact of model distance metrics and complexity indicator values*: As stated in Section V-B, model distances  $\hat{d}(\theta_m, \theta_j)$  are computed to approximate distribution divergences. Four distance metrics (i.e.,  $\ell_1/\ell_2$ -norm, inner product, and cosine similarity) are evaluated for two heterogeneous distributions, respectively. As shown in Fig. 6(a),  $\ell_2$ -norm achieves the lowest validation BER under two types of heterogeneity, and thus  $\ell_2$ -norm is selected as the desired metric. In addition, complexity indicator  $C$  in objective (7) is varied from 0.2 to 0.8, which means the importance of quantity-aware term increases. As shown in 6(b), for Hetero-1,  $C$  should be between 0.6 and 0.8, while for Hetero-2, low BERs are achieved with  $C$  from 0.2 to 0.8. Therefore, this complexity indicator is set to 0.6.

3) *Impact of collaboration graph directivity and visualization of learned graphs*: Since the collaboration relationship is not necessarily reciprocal, a directed collaboration graph is learned in GraphRx. To validate this point, the performance of GraphRx with directed and undirected graphs are compared in Fig. 7. Note that graph weight matrix  $\mathbf{W}$  must be symmetric in an undirected graph, which is added as a constraint in solving

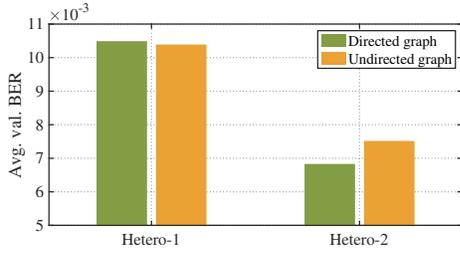
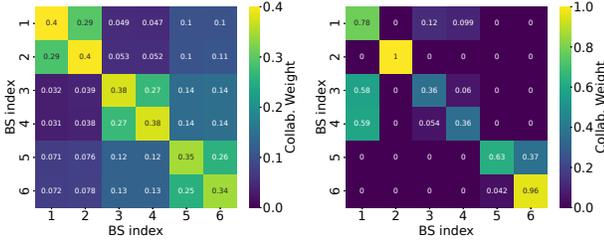


Fig. 7. Average validation BER with directed/undirected graphs.



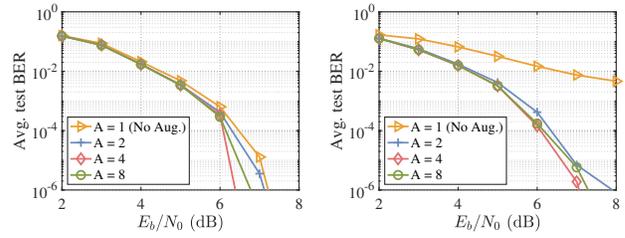
(a) Hetero-1 (b) Hetero-2  
Fig. 8. Visualization of directed collaboration graphs.

(10)-(11). For Hetero-1 where all BSs hold the same data quantity, graph directivity has little impact on validation BER, while for Hetero-2, the BER drops by 9.13% for Hetero-2 if a directed graph is learned instead of an undirected one. This is because in the latter case each BS has various data quantities: contribution from BSs with large data quantities to those with smaller quantities is more significant than contribution in the opposite direction, which is also validated by the following graph visualization.

The learned collaboration weights  $\mathbf{W}$  of directed graphs with GraphRx are shown in Fig. 8 for two heterogeneous distributions. Specifically,  $w_{mj}$  is the value at the  $m$ -th row and the  $j$ -th column, which indicates how much BS  $m$ 's model  $\theta_m$  learns from BS  $j$ 's model  $\theta_j$ . In Fig. 8(a), there exist three pairs of BSs (1,2), (3,4), (5,6) exhibiting strong collaboration weights. This is well aligned with their true data characteristics in Table I. In Fig. 8(b), BS 2 does not need to collaborate with others based on the learned graph, as it holds an enough quantity of data and may not benefit from collaboration. BS 5 benefits much more by learning from BS 6 than that in the reverse direction. Also, strong uni-directional collaboration is established in each BS pair of (3,1) and (4,1).

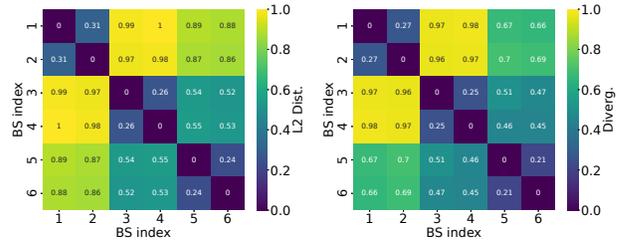
4) *Validation of data augmentation mechanism:* The evaluation results of data augmentation is presented for Hetero-1 in both interference-free and interference-included cases. Specifically, through extensive trials, a quantity of around 6400 unaugmented TTIs is required at each BS to guarantee convergence. As the training pilot overhead is controlled as low as 0.1%, one training pilot frame is inserted among 1,000 transmitted OFDM frames. In this case, it requires around 1-2 hours (collaboration period) for each BS to collect 6,400 TTIs. As previously stated in Sec. IV, during the collaboration period, the currently deployed neural receiver operates normally.

To achieve better performance without increasing pilot overhead or storage cost, data augmentation is conducted for

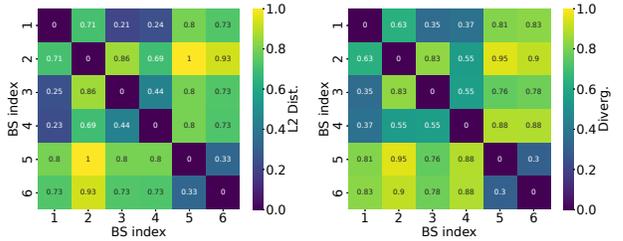


(a) Without interference (b) With inter-cell interference

Fig. 9. Coded BER of GraphRx tested under various augmentation times.



(a)  $\ell_2$  model distance (Hetero-1) (b)  $\mathcal{H}\Delta\mathcal{H}$ -divergence (Hetero-1)



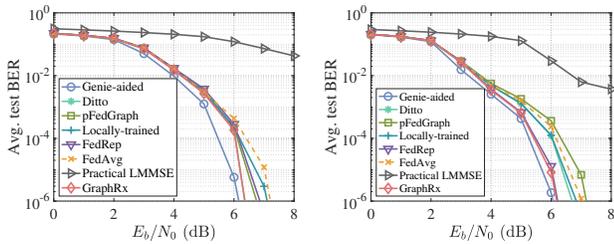
(c)  $\ell_2$  model distance (Hetero-2) (d)  $\mathcal{H}\Delta\mathcal{H}$ -divergence (Hetero-2)

Fig. 10. Correlations between model distance and estimated divergence.

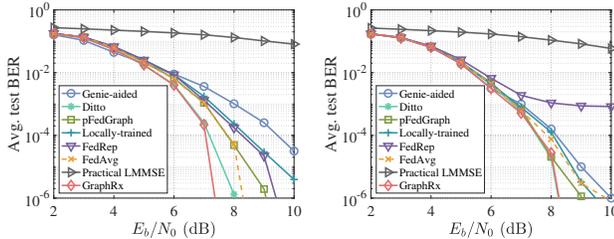
each local dataset. As shown in Fig. 9(a), when augmentation time  $A = 4$ , the average test BER of all neural receivers achieves 1 dB SNR gain at  $\text{BER} = 10^{-5}$  compared with that in the unaugmented case. However, when  $A$  increases to 8, the BER performance drops as intensive augmentation may cause overfitting. Similar patterns can also be observed in Fig. 9(b) where inter-cell interference is considered, and a significant gain of around 3 dB is achieved at  $\text{BER} = 10^{-3}$  compared with the unaugmented case. Therefore, in GraphRx, data augmentation with  $A = 4$  is adopted to enrich each local dataset. This reduces the pilot overhead by 75% compared with collecting four times the current quantity of data.

### C. Effectiveness of Divergence Approximation

Due to the unavailability of theoretical bound in practice,  $\ell_2$  model distance is used to approximate pairwise divergence  $d_{mj}$ . In domain adaptation tasks,  $\lambda_{mj}$  is generally small and thus can be ignored [28]. Following the common method in [27], [28],  $\mathcal{H}\Delta\mathcal{H}$ -divergence is empirically estimated by training a non-linear classifier to discriminate between any pair of local datasets. There exists a strong positive correlation between each BS pair's model distance and distribution divergence under both Hetero-1 and Hetero-2. Taking BS 1 shown in Fig. 10(a) and 10(b) as an example, BS pair (1,2) has both the smallest distance and divergence at the same time, and



(a) Hetero-1  
 (b) Hetero-2  
 Fig. 11. Coded BER tested without inter-cell interference.



(a) Hetero-1  
 (b) Hetero-5  
 Fig. 12. Coded BER tested with inter-cell interference.

BS 1 exhibits large distances and divergences simultaneously from BSs 3 and 4. Such positive correlations are observed by comparing each row of the distance matrix and divergence matrix in Fig. 10.

#### D. Comparison with Baselines

The test performance of GraphRx is evaluated and compared with other baseline receivers. For neural receivers, data augmentation with  $A = 4$  is adopted for all the algorithms for fair comparison. The SNR gain is measured at  $\text{BER} = 10^{-5}$  unless stated otherwise.

First, we consider the scenario without inter-cell interference as shown in Fig. 11 for Hetero-1 and Hetero-2. As observed in both figures, the genie-aided LMMSE receiver serves as the performance upper bound in the interference-free case, outperforming GraphRx by 0.3 dB for Hetero-1 and 0.2 dB for Hetero-2. Our GraphRx achieves the best BER performance among all the other baselines under these two heterogeneous distributions. Specifically, for Hetero-1, GraphRx exceeds both pFedGraph and FedRep by around 0.4 dB and it outperforms FedAvg by 0.9 dB. Note that the performance of Ditto is very close to GraphRx for Hetero-1, while Ditto performs 0.4 dB worse than GraphRx for Hetero-2 shown in Fig. 11(b). In addition, under both Hetero-1 and Hetero-2, FedAvg is inferior to local learning by around 0.2 dB, which reflects that learning a unified global model is not optimal for improving local performance when data are heterogeneously distributed. It is also worth noting that all the methods based on neural receivers can outperform the practical LMMSE receiver by a significant margin of over 4 dB.

Next, we consider the scenario with inter-cell interference shown in Fig. 12. Under both Hetero-1 and Hetero-2, the genie-aided LMMSE receiver is no longer the performance upper bound since it does not have any knowledge of the interfering channels. Under Hetero-1 illustrated in Fig. 12(a),

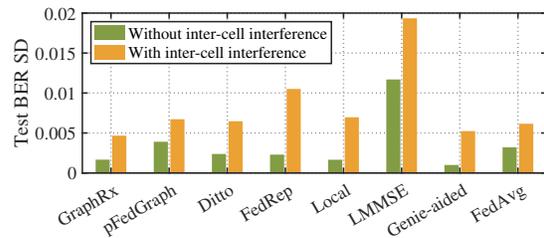


Fig. 13. Standard deviation of coded BER tested for Hetero-2.

GraphRx achieves notable performance gains over all the listed baselines. Specifically, it outperforms three other PFL algorithms (i.e., Ditto, pFedGraph, and FedRep) by 0.5 dB, 1.5 dB, and 2.1 dB, respectively. Interestingly, the performance of FedAvg is 1.6 dB better than that of local learning, which suggests that even a non-personalized aggregate model can achieve a substantial gain when inter-cell interference exists. Under Hetero-2 shown in Fig. 12(b), GraphRx performs equally well as Ditto, but outperforms all other baselines. Two additional insights are drawn: 1) GraphRx is robust to interference, while the performance of FedRep varies significantly; 2) when interference exists, collaboration learning seems more challenging under Hetero-2 than Hetero-1, as Hetero-2 has a higher heterogeneity level in both local distributions and data quantities than Hetero-1. This issue is subject to future research.

Finally, the fairness among six BSs is presented in Fig. 13, which is quantified by the standard deviation (SD) of the test BERs of all neural receivers. The SD is evaluated at  $\text{SNR} = 5\text{dB}$  for Hetero-2 that has a higher heterogeneity level. As shown in Fig. 13, GraphRx achieves the lowest BER SD whether interference exists or not, which means the best fairness can be guaranteed among all collaborators. In addition, the BER SD values of all the methods become larger with inter-cell interference.

#### VIII. CONCLUSION

A multi-cell collaborative learning scheme named GraphRx was developed in this paper. With a weighted graph capturing collaboration relationships among BSs, GraphRx was formulated as a personalized federated learning problem. To solve this problem, the personalized models were updated locally at each BS and the collaboration graph was optimized centrally at the server by minimizing an approximate generalization bound, and these two steps were performed alternatively. Data augmentation techniques were further incorporated into GraphRx to reduce overhead of training pilot signals. GraphRx achieved significant performance gains over the baselines.

GraphRx is designed based on an existing neural receiver architecture, but this architecture itself demands more study, particularly in two aspects. First, an architecture that does not need DMRS signals is desired. Second, how to support MIMO communications by a neural receiver requires more exploration. Moreover, further research is needed to handle scenarios with both inter-cell interference and high heterogeneity among cells. The authors have provided public access to their code at <https://github.com/tesiaawang/GraphRx>.

## REFERENCES

- [1] M. Honkala, D. Korpi, and J. M. Huttunen, "DeepRx: Fully convolutional deep learning receiver," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 6, pp. 3925–3940, 2021.
- [2] P. Jiang, T. Wang, B. Han, X. Gao, J. Zhang, C.-K. Wen, S. Jin, and G. Y. Li, "AI-aided online adaptive OFDM receiver: Design and experimental results," *IEEE Trans. Wirel. Commun.*, vol. 20, no. 11, pp. 7655–7668, 2021.
- [3] M. B. Fischer, S. Dörner, F. Krieg, S. Cammerer, and S. ten Brink, "Adaptive NN-based OFDM receivers: Computational complexity vs. achievable performance," in *IEEE Asilomar Conf. Signals Syst. Comput.*, 2022, pp. 194–199.
- [4] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to demodulate from few pilots via offline and online meta-learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 226–239, 2021.
- [5] M. B. Fischer, S. Dörner, S. Cammerer, T. Shimizu, H. Lu, and S. Ten Brink, "Adaptive neural network-based OFDM receivers," in *IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2022, pp. 1–5.
- [6] M. B. Mashhadi, N. Shlezinger, Y. C. Eldar, and D. Gündüz, "Fedrec: Federated learning of universal receivers over fading channels," in *2021 IEEE Stat. Signal Process. Workshop (SSP)*. IEEE, 2021, pp. 576–580.
- [7] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, 2020.
- [8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artif. Intell. Stat.*, 2017, pp. 1273–1282.
- [9] R. Ye, Z. Ni, F. Wu, S. Chen, and Y. Wang, "Personalized federated learning with inferred collaboration graphs," in *Int. Conf. Mach. Learn. (ICML)*, 2023.
- [10] F. A. Aoudia and J. Hoydis, "End-to-end learning for ofdm: From neural receivers to pilotless communication," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 2, pp. 1049–1063, 2021.
- [11] —, "Waveform learning for next-generation wireless communication systems," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3804–3817, 2022.
- [12] J. Pihlajasalo, D. Korpi, M. Honkala, J. M. Huttunen, T. Riihonen, J. Talvitie, A. Brihuega, M. A. Uusitalo, and M. Valkama, "Deep learning ofdm receivers for improved power efficiency and coverage," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 8, pp. 5518–5535, 2023.
- [13] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wirel. Commun. Lett.*, vol. 7, no. 1, pp. 114–117, 2017.
- [14] H. Ye, L. Liang, G. Y. Li, and B.-H. Juang, "Deep learning-based end-to-end wireless communication systems with conditional gans as unknown channels," *IEEE Trans. Wirel. Commun.*, vol. 19, no. 5, pp. 3133–3143, 2020.
- [15] H. Ye, G. Y. Li, and B.-H. Juang, "Deep learning based end-to-end wireless communication systems without pilots," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 702–714, 2021.
- [16] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, 2018.
- [17] Z. Zhao, M. C. Vuran, F. Guo, and S. D. Scott, "Deep-waveform: A learned ofdm receiver based on deep complex-valued convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2407–2420, 2021.
- [18] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.
- [19] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9587–9603, 2022.
- [20] W. Bao, H. Wang, J. Wu, and J. He, "Optimizing the collaboration structure in cross-silo federated learning," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2023, pp. 1718–1736.
- [21] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 6357–6368.
- [22] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 2089–2099.
- [23] T. Wang, S. Wang, Y. G. Li, and X. Wang, "Collaborative learning for less online retraining of neural receivers," in *Proc. IEEE Int. Workshop Machine Learn. Signal Process. (MLSP)*, 2024.
- [24] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, pp. 19 586–19 597, 2020.
- [25] 3rd Generation Partnership Project (3GPP), "NR;physical channels and modulation," 3GPP, TS 38.211, 2024, v18.3.0.
- [26] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 2, pp. 1376–1411, 2023.
- [27] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, pp. 151–175, 2010.
- [28] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 20, 2007.
- [29] V. Zantedeschi, A. Bellet, and M. Tommasi, "Fully decentralized joint learning of personalized models and collaboration graphs," in *Proc. Int. Conf. Artif. Intell. Stat.*. PMLR, 2020, pp. 864–874.
- [30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [31] 3rd Generation Partnership Project (3GPP), "Study on channel model for frequencies from 0.5 to 100 GHz," 3GPP, TS 38.901, 2024, v17.1.0.
- [32] —, "NR; requirements for support of radio resource management," 3GPP, TS 38.133, 2024, v18.6.0.