Distributed Downlink Precoding for Cell-Free Massive MIMO: A Quasi-Neural Network Approach

Weijie Dai, Rui Wang, Junkai Liu, and Yi Jiang, *Member, IEEE*

***Abstract*—This paper proposes a novel downlink precoding method for a cell-free massive multiple-input multiple-output (CF-mMIMO) network, requiring no channel state information sharing between the access points via fronthaul links. By drawing analogies between a CF-mMIMO network and an artificial neural network, the proposed algorithm borrows the idea of backpropagation to train the precoders and the combiners through over-the-air ping-pong signaling between the access points and user equipments. It utilizes manifolds optimization to meet the per-AP power constraint and is named as distributed quasi-neural network precoding on manifold (DQNPM). The DQNPM algorithm can accommodate a large category of objective functions for fully distributed implementation. Numerical simulations show that our method outperforms the state-of-the-art approaches, and is robust against pilot contamination.**

***Index Terms*—cell-free massive MIMO, distributed precoding, quasi-neural network, Riemannian manifold**

I. Introduction

THE cell-free massive multiple-input and multiple-output (CF-mMIMO) network [2], [3] is a recently emerged physical layer technology, which combines distributed access points (APs) to achieve the high throughput of massive MIMO [4] and ensure more uniform coverage. Hence it is one of the promising technologies for the future generation of wireless communication. In a CF-mMIMO network, the APs collaboratively serve all user equipments (UEs) [5], or specific clusters of UEs determined by some AP-UE pairing schemes, such as the UE-centric allocation [6].

To fully leverage the capacity of the CF-mMIMO network, extensive research efforts have been made in recent years, including the precoding and combining with different level of cooperations between the APs [7], [8], [9], pilot assignment with a limited number of orthogonal pilots [10], [11], and power allocation [6], [12].

The cooperative precoding and combining design in a CF-mMIMO network can be achieved by centralized or distributed approaches. The centralized approaches require the aggregation of the CSI of the distributed APs to the central processing unit (CPU) via the fronthaul links so that the precoders can be optimized in a central manner. Along this vein of work, the centralized zero-forcing (ZF) precoding [13], [14] and the minimum mean-square-error (MMSE) combining [15] are adopted to implement in a CF-mMIMO network at different levels of cooperation [16]. Due to the high dimensionality of the aggregated channels, however, the centralized methods entails a large overhead of CSI aggregation and are computationally complex. Moreover, the centralized precoding algorithms are hard to implement due to the limited fronthaul capacity in a centralized radio access network (CRAN) [17] and the fronthaul delay in an IP-radio access network (IP-RAN) [18]. To bypass these problems, the distributed approaches were proposed, including the local ZF downlink precoding [13], signal-to-leakage-and-noise ratio (SLNR) precoding [19] and MMSE [15] precoding. These algorithms only require the local CSI, and therefore incurring no CSI or data exchanges over the fronthaul links. But they significantly underperform compared with the centralized ones. Another way is the bi-directional training via the over-the-air (OTA) pilot transmissions between the APs and the UEs, which exchanges the CSI implicitly and distributively to optimizes the precoders. This method appears to be able to achieve a better performance than the previous methods [15], [13], [19], but it still sees considerable performance loss compared with a centralized benchmark.

This paper focuses on the distributed cooperative precoding and combing in a downlink CF-mMIMO network. We adopt the idea of “quasi-neural network” (Quasi-NN), which was originally proposed for distributed beamforming in a MIMO relay network [20] and was used for distributed combining of the APs in the uplink scenario of a CF network [21]. The Quasi-NN is similar to an artificial neural network (ANN) in its layer structure that directly models the hierarchical feature of a system, such as the layer structure of relay networks [20], CF-mMIMO networks [21], and the signal processing structure in spectrum estimations [22]. Meanwhile, it differs from an ANN as it is model-based and optimized in real time. It benefits from the ability of optimizing complicated systems, parallel and distributed implementations, low computational complexity, and even pruning techniques [22].

We model the CF-mMIMO network in the downlink transmission as a Quasi-NN by drawing its analogies to an ANN. By using stochastic gradient descent with OTA signals between the APs and the UEs, which is the essential idea of the celebrated back-propagation (BP) algorithm [23], the coefficients of the Quasi-NN, i.e., the precoding matrices of the APs and the combiners of the UEs, can be distributed optimized. To constrain the transmitting power of the APs, we propose to optimize the precoders over a Riemannian manifold.

The main contributions of this paper can be concluded as follows:

The authors are with the Key Laboratory for Information Science of Electromagnetic Waves (MoE), Dept. of Communication Science and Engineering, School of Information Science and Technologies, Fudan University, Shanghai 200438, China. *Corresponding author: Yi Jiang, yijiang@fudan.edu.cn*

Part of this work was presented at the 2023 IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Shanghai, China [1].

- We propose the distributed quasi-neural network precoding on manifolds (DQNPM) algorithm, which uses OTA signaling to optimize the precoding matrices of the APs and the combiners of the UEs in a distributed manner without fronthaul data sharing. It is shown to enjoy better convergence performance than the approach of [21].
- We extend the DQNPM algorithm into multi-stream per UE communication scenario.
- We delineate a category objective functions the Quasi-NN can accommodate for fully-distributed optimization, including minimization of the weighted mean squared error (MMSE), maximization of the weighted sum rate (MWSR), and maximization of the proportional fairness (MPF).

In the remainder of this paper, we first introduce the network model of a CF-mMIMO and draw its analogies to an ANN in Section II. The DQNPM algorithm and its implementation under the MWSR and MPF criteria are elaborated in Section III. In Section IV, we discuss what kind of objective functions can be distributively optimized by the DQNPM algorithm and extend it into a multi-stream scenario. The proposed methods are verified by numerical simulations in Section V. The conclusions are given in Section VI.

Notation: Small and capital boldface variables denote vectors and matrices. $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^*$ denote the transpose, conjugate transpose, and conjugate operator. $\Re\{\cdot\}$ and $\mathbb{E}[\cdot]$ are the real part and expectation operator. $|\mathcal{S}|$ stands for the cardinality of the set $\mathcal{S}$. $\mathbf{I}$ stands for an identity matrix. $\text{vec}(\mathbf{X})$ denotes the vectorization operation by stacking the columns of $\mathbf{X}$ into one. $\otimes$ stands for the the Kronecker product, and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

II. Problem Formulation and Modeling of Downlink CF-mMIMO Network by a Quasi-NN

Consider a downlink CF-mMIMO network as shown in Fig. 1, where a group of APs $\mathcal{L} = \{1, \cdots, L\}$, each equipped with M_t transmitting antennas, serve a set of UEs $\mathcal{K} = \{1, \cdots, K\}$, each with M_r receiving antennas. The key idea of this paper is to model such a network by a Quasi-NN and present a distributed algorithm for optimizing the network, which is inspired by the analogies between the Quasi-NN and a conventional ANN.

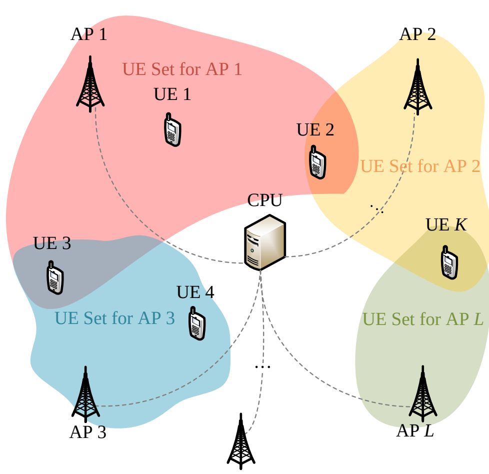


Fig. 1: Downlink CF-mMIMO System

A. Signal Model and Problem Formulation

Assume that the network works in a time-division-duplex (TDD) mode and the APs transmit the data stream s_k to the UE k. Based on some predefined AP-to-UE pairing, the CPU allocates a subset of the data streams $\mathcal{S}_l \subseteq \mathcal{S} = \{s_1, \cdots, s_K\}$ to AP l. Denote $\mathbf{s}_l \in \mathbb{C}^{N_l \times 1}$ as the vector of the signals allocated to AP l, with $N_l = |\mathcal{S}_l|$ being the number of the UEs served by AP l. Applied with the precoding matrix $\mathbf{P}_l = [\mathbf{p}_{l,1}, \cdots, \mathbf{p}_{l,N_l}] \in \mathbb{C}^{M_t \times N_l}$, it is precoded into

$$\mathbf{x}_l = \mathbf{P}_l \mathbf{s}_l \in \mathbb{C}^{M_t \times 1} \tag{1}$$

before being transmitted from the AP's antenna array. Without loss of generality, it is assumed that $\mathbb{E}[\mathbf{s}_l \mathbf{s}_l^H] = \mathbf{I}$ for any l.

The signal received by UE k is

$$\mathbf{y}_k = \sum_{l=1}^{L} \mathbf{H}_{k,l} \mathbf{x}_l + \mathbf{n}_k \in \mathbb{C}^{M_r \times 1}, \tag{2}$$

with $\mathbf{n}_k \sim \mathcal{CN}(0, \sigma_k^2 \mathbf{I})$ denoting the additive white Gaussian noise (AWGN) and $\mathbf{H}_{k,l} \in \mathbb{C}^{M_r \times M_t}$ denoting the downlink channel between AP l and UE k. Then, UE k applies combiner $\mathbf{w}_k \in \mathbb{C}^{M_r \times 1}$ to recover s_k as:

$$\hat{s}_k = \mathbf{w}_k^H \mathbf{y}_k. \tag{3}$$

This paper focuses on the distributed optimization of the precoding matrices $\{\mathbf{P}_l\}_{l \in \mathcal{L}}$ and the UE combining weights $\{\mathbf{w}_k\}_{k \in \mathcal{K}}$ according to some predefined criterion J, subject to the power constraint per-AP. That is

$$\begin{aligned} \min_{\{\mathbf{P}_l\}_{l \in \mathcal{L}}, \{\mathbf{w}_k\}_{k \in \mathcal{K}}} \quad & J\left(\{\mathbf{P}_l\}_{l \in \mathcal{L}}, \{\mathbf{w}_k\}_{k \in \mathcal{K}}\right) \\ \text{s.t.} \quad & \|\mathbf{P}_l\|_F^2 \leq 1, \quad l = 1, \cdots, L. \end{aligned} \tag{4}$$

The cost functions $J(\{\mathbf{P}_l\}_{l \in \mathcal{L}}, \{\mathbf{w}_k\}_{k \in \mathcal{K}})$ can be in various forms. The following are probably two most interesting examples.

For the MWSR criterion, maximizing the output SNR amounts to minimizing the system MSE [24, Equation (13)] because

$$\text{SNR} = \frac{1}{\text{MSE}} - 1, \tag{5}$$

which holds if the MMSE receiver

$$\mathbf{w}_k = \left(\mathbf{Y}_k \mathbf{Y}_k^H\right)^{-1} \mathbf{Y}_k \mathbf{s}_k^*, \tag{6}$$

is employed. Here, $\mathbf{Y}_k \in \mathbb{C}^{M_r \times \tau}$ and $\mathbf{s}_k \in \mathbb{C}^{\tau \times 1}$ denote the received signal and pilot sequence for UE k.

Hence, the weighted sum-rate of the CF-mMIMO network $R = \sum_{k=1}^{K} \omega_k \log(1 + \text{SNR}_k) = -\sum_{k=1}^{K} \omega_k \log(\text{MSE}_k)$, which can be approximated as:

$$R = -\sum_{k=1}^{K} \omega_k \log\left(\frac{1}{\tau} \sum_{i=1}^{\tau} |\hat{s}_k(i) - s_k(i)|^2\right), \tag{7}$$

This article has been accepted for publication in IEEE Transactions on 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/TCOMM.2024.3443735

where ω_k denotes some prescribed weight for UE k. The approximation $\mathrm{MSE}_k \simeq \frac{1}{\tau}\sum_{i=1}^{\tau}|\hat{s}_k(i)-s_k(i)|^2$ is utilized, and it is asymptotically accurate as $\tau \to \infty$.

Hence for the MWSR criterion, the adopted cost function is

$$J \triangleq \sum_{k=1}^{K}\omega_k \log\left(\frac{1}{\tau}\sum_{i=1}^{\tau}|\hat{s}_k(i)-s_k(i)|^2\right). \tag{8}$$

For the MPF criterion, the proportional fairness (PF) can be similarly approximated as

$$\mathrm{PF} = \sum_{k=1}^{K}\log\left(-\log\left(\frac{1}{\tau}\sum_{i=1}^{\tau}|\hat{s}_k(i)-s_k(i)|^2\right)\right). \tag{9}$$

Hence for the MPF criterion, the adopted cost function is

$$J = -\sum_{k=1}^{K}\log\left(-\log\left(\frac{1}{\tau}\sum_{i=1}^{\tau}|\hat{s}_k(i)-s_k(i)|^2\right)\right). \tag{10}$$

Note that the term $\frac{1}{\tau}\sum_{i=1}^{\tau}|\hat{s}_k(i)-s_k(i)|^2$ is guaranteed to be less than 1 when the MMSE receiver is used, since $\mathbb{E}[|s_k(i)|^2]=1$ for any k. Hence, (10) is always well-defined.

For both the MWSR and MPF criterion, the optimal solution to $\mathbf{w}_k$ is the MMSE one (6). Indeed, in a network where all the nodes are subject to Gaussian noise, it is hard to conceive a criterion under which a MMSE combiner is not the preferred one. Further note that all the UEs' MMSE combiners can be optimized separately, which is in contrast to the APs' precoders – when AP l changes its precoder $\mathbf{P}_l$, the optimal precoders $\mathbf{P}_{l'}$ for $l' \neq l$ will be affected accordingly.

Given the MMSE combiners, denote the resultant MSEs by $e_k = |\hat{s}_k - s_k|^2$ or $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2]$ for $k = 1, \ldots, K$, each of which is a function of $\mathbf{P}_l, l = 1, \ldots, L$; thus, the original cost function $J\left(\{\mathbf{P}_l\}_{l\in\mathcal{L}}, \{\mathbf{w}_k\}_{k\in\mathcal{K}}\right)$ can be converted into, with a slight abuse of notation, $J(e_1(\{\mathbf{P}_l\}_{l\in\mathcal{L}}), \cdots, e_K(\{\mathbf{P}_l\}_{l\in\mathcal{L}}))$. Here the notation $e_k(\{\mathbf{P}_l\}_{l\in\mathcal{L}})$ is to emphasize that e_k is dependent on all the precoding matrices.

Hence the original problem (4) can be converted into

$$\begin{aligned} \min_{\{\mathbf{P}_l\}_{l\in\mathcal{L}}} \quad & J\left(e_1(\{\mathbf{P}_l\}_{l\in\mathcal{L}}), \cdots, e_K(\{\mathbf{P}_l\}_{l\in\mathcal{L}})\right) \\ \text{s.t.} \quad & \|\mathbf{P}_l\|_F^2 \le 1, \quad l = 1, \cdots, L. \end{aligned} \tag{11}$$

However, obtaining a distributed solution for this problem is still troublesome due to its non-convexity and fronthaul limitations. One interesting observation is that the layered structure of a CF-mMIMO system makes it similar to an ANN, and we can model it as a Quasi-NN as explained in the following. In Section III, we deduce a distributed algorithm for solving problem (11) with the MWSR and MPF criteria, which is based on the notion of Quasi-NN.

B. The Quasi-NN Modeling of Downlink CF-mMIMO Network

The layered topology of the downlink CF-mMIMO network is shown in the upper subplot of Fig. 2, where different colors of the links indicate the AP-to-UE pairing as the same as those in Fig. 1. We can model this topology into a four-layer Quasi-NN in the lower subplot of Fig. 2 based on the following similarities and differences.

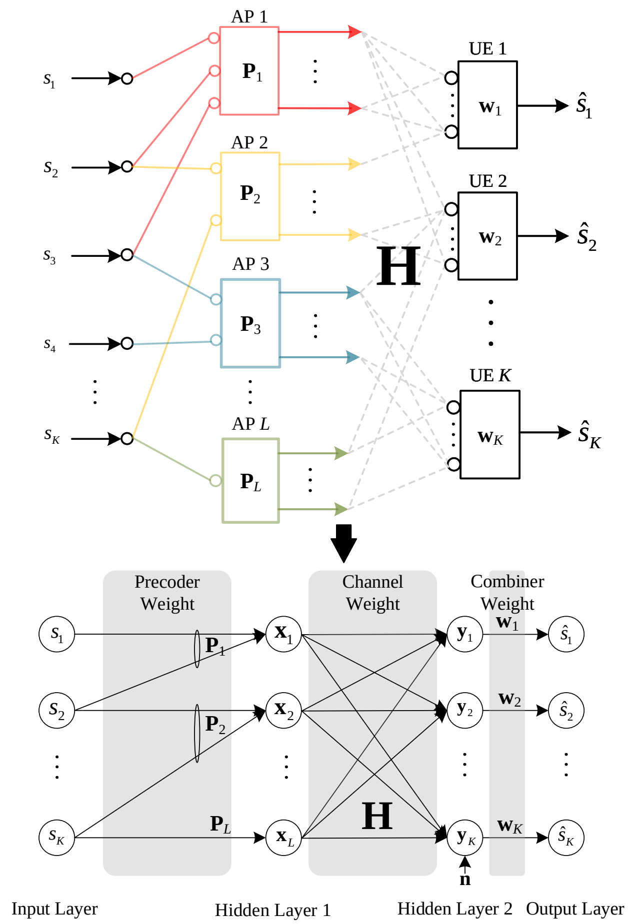


Fig. 2: Modeling a Cell-free Network as a Quasi-NN

The CF-mMIMO network is analogous to an ANN in the following aspects:

- The input streams are like the input layer of an ANN; while the combined streams $\hat{s}_k$'s are like the output layer.
- The transmitting antennas of the APs and the receiving antennas at the UE are similar to the neurons in the ANN's hidden layers.
- The precoding matrices $\mathbf{P}_l$'s, the channel matrices $\mathbf{H}_{k,l}$'s, and the combining weights $\mathbf{w}_k$'s are like the connection weights of a four-layer ANN.

One should note that the Quasi-NN modeling of the CF-mMIMO network also differs from an ANN in some important aspects:

- In the Quasi-NN, the channel weights are deterministic and unknown, while all network weights are known and adjustable in an ANN.
- The precoding weights in the Quasi-NN follow the transmitting power constraint; while the network weights in an ANN are usually unconstrained.
- The pilot transmission in the Quasi-NN is contaminated by AWGN, while the data processing is typically noise-free in an ANN.
- Data and weights in the Quasi-NN are complex-valued, while they are typically real in an ANN.

This article has been accepted for publication in IEEE Transactions on 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/TCOMM.2024.3443735

Despite the aforementioned differences, we can still borrow the idea of the BP algorithm used for neural network training to solve the network optimization problem (11). Specifically, we let the APs to transmit pilot sequences in the downlink and the UEs to transmit the gradients in the uplink to distributively optimize problem (11), while assuming that no CSI is available at the network nodes, as is detailed in the next section.

III. The Distributed Quasi-Neural Network Precoding on Manifold

To employ the BP algorithm to optimize the CF-mMIMO network, we need further to reformulate (11) into an unconstrained optimization problem. Indeed, in our previous work on the optimization of the CF-mMIMO network in the uplink mode [21], we introduce power control factors to reformulate the constrained problem into an unconstrained one. That approach can be easily altered to solve problem (11) and the resultant method is referred to as the distributed quasi-neural network precoding (DQNP) algorithm, whose details are relegated to Appendix B.

In this section, we propose a new approach, named as the DQNPM algorithm, which solves problem (11) over Riemannian manifolds to achieve better convergence performance than the DQNP.

A. Precoding Problem Representation on Manifolds

To meet the power constraint in (11), i.e., $\|\mathbf{P}_l\|_F^2 = \sum_{i=1}^{N_l} \operatorname{tr}\left(\mathbf{p}_{l,i}\mathbf{p}_{l,i}^H\right) \leq 1$ with $\mathbf{p}_{l,i}$ the i-th column of $\mathbf{P}_l$, we first introduce the slack variables $\boldsymbol{\theta}_l \in \mathbb{C}^{N_l\times 1}$ to have the slacked precoders

$$\tilde{\mathbf{P}}_l = \begin{bmatrix} \boldsymbol{\theta}_l^H \\ \mathbf{P}_l \end{bmatrix}, \quad l = 1, \cdots, L \tag{12}$$

with the equality constraint $\|\tilde{\mathbf{P}}_l\|_F^2 = 1$.

With $\mathbf{A} = [\mathbf{0} \;\; \mathbf{I}_{M_t}] \in \mathbb{R}^{M_t\times(M_t+1)}$, the transmitted signal in (1) is

$$\mathbf{x}_l = \mathbf{A}\tilde{\mathbf{P}}_l\mathbf{s}_l. \tag{13}$$

Therefore, the transmitting power constraint $\|\mathbf{A}\tilde{\mathbf{P}}_l\|_F^2 \leq 1$ holds automatically.

Consider the set $\mathcal{M}_l = \{\tilde{\mathbf{p}}_l \mid \tilde{\mathbf{p}}_l^H\tilde{\mathbf{p}}_l = 1\}$, where $\tilde{\mathbf{p}}_l = \operatorname{vec}(\tilde{\mathbf{P}}_l) \in \mathbb{C}^{(M_t+1)N_l}$, and define the inner product

$$g_{\tilde{\mathbf{p}}_l}^{\mathcal{M}_l}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{1}{2}\left(\boldsymbol{\phi}^H\boldsymbol{\psi} + \boldsymbol{\psi}^H\boldsymbol{\phi}\right), \tag{14}$$

of two vectors $\boldsymbol{\phi}, \boldsymbol{\psi}$ on the tangent space $T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l$, which is the set of the tangent vectors of all the curves passing through $\tilde{\mathbf{p}}_l$ [25]. $\mathcal{M}_l$ is a complex sphere and forms a Riemannian manifold [26].

Hence, we can transform (11) to the manifold optimization as the following $\tilde{\mathbf{p}}_l$ on manifold $\mathcal{M}_l$:

$$\begin{aligned} \min_{\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}} \quad & J(e_1(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}), \cdots, e_K(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}})) \\ \text{s.t.} \quad & \tilde{\mathbf{p}}_l \in \mathcal{M}_l, \quad (l = 1, \cdots, L). \end{aligned} \tag{15}$$

B. The Distributed Quasi-NN Precoding on Manifold

We adopt the first-order Riemannian gradient descent method [26] to optimize $\tilde{\mathbf{p}}_l$. The basic idea is to update the precoder $\tilde{\mathbf{p}}_l$ along the Riemannian gradient on the tangent space $T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l$ before retracting it back to the manifold. To acquire the Riemannian gradient, we first establish the following result on Euclidean gradient.

Proposition 1. *For the Quasi-NN as shown in Fig. 2, the gradient of $J(e_1(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}), \cdots, e_K(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}))$ with respect to $\tilde{\mathbf{p}}_l$ is*

$$\frac{\partial J}{\partial \tilde{\mathbf{p}}_l^*} = (\mathbf{s}_l^* \otimes \mathbf{A}^T)\frac{\partial J}{\partial \mathbf{x}_l^*}. \tag{16}$$

with

$$\frac{\partial J}{\partial \mathbf{x}_l^*} = \sum_{k=1}^{K} \mathbf{H}_{k,l}^H \mathbf{w}_k(\hat{s}_k - s_k)\frac{\partial J}{\partial e_k}. \tag{17}$$

Proof. We first consider the case $e_k = |\hat{s}_k - s_k|^2$. According to (3) and the chain rule, we can derive

$$\frac{\partial e_k}{\partial \mathbf{y}_k^*} = \mathbf{w}_k(\hat{s}_k - s_k). \tag{18}$$

By (2), we have

$$\frac{\partial \mathbf{y}_k^H}{\partial \mathbf{x}_l^*} = \mathbf{H}_{k,l}^H, \quad \frac{\partial \mathbf{y}_k^T}{\partial \mathbf{x}_l^*} = 0. \tag{19}$$

According to the chain rule,

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{x}_l^*} &= \sum_{k=1}^{K}\left(\frac{\partial \mathbf{y}_k^H}{\partial \mathbf{x}_l^*}\cdot\frac{\partial e_k}{\partial \mathbf{y}_k^*}\cdot\frac{\partial J}{\partial e_k} + \frac{\partial \mathbf{y}_k^T}{\partial \mathbf{x}_l^*}\cdot\frac{\partial e_k}{\partial \mathbf{y}_k}\cdot\frac{\partial J}{\partial e_k}\right) \\ &= \sum_{k=1}^{K} \mathbf{H}_{k,l}^H \mathbf{w}_k(\hat{s}_k - s_k)\frac{\partial J}{\partial e_k}. \end{aligned} \tag{20}$$

By the formula $\operatorname{vec}(\mathbf{CXB}) = (\mathbf{B}^T \otimes \mathbf{C})\operatorname{vec}(\mathbf{X})$, we have

$$\frac{\partial J}{\partial \tilde{\mathbf{p}}_l^*} = \operatorname{vec}\left(\mathbf{A}^T\frac{\partial J}{\partial \mathbf{x}_l^*}\mathbf{s}_l^H\right) = (\mathbf{s}_l^* \otimes \mathbf{A}^T)\frac{\partial J}{\partial \mathbf{x}_l^*}. \tag{21}$$

The result for case $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2] \simeq \frac{1}{\tau}\sum_{i=1}^{\tau} |\hat{s}_k(i) - s_k(i)|^2$ can be easily extended by applying the above results into every $|\hat{s}_k(i) - s_k(i)|, (i = 1, \cdots, \tau)$. That is

$$\frac{\partial e_k}{\partial \mathbf{y}_k^*(i)} = \frac{1}{\tau}\mathbf{w}_k(\hat{s}_k(i) - s_k(i)) \quad (i = 1, \cdots, \tau). \tag{22}$$

$\square$

Note that the partial derivative $\frac{\partial J}{\partial e_k}$ varies for different cost functions. For the MWSR criterion (8),

$$\frac{\partial J}{\partial e_k} = \frac{\omega_k}{e_k}. \tag{23}$$

For the MPF criterion (10),

$$\frac{\partial J}{\partial e_k} = -\frac{1}{e_k \log(e_k)}. \tag{24}$$

Proposition 1 provides insights on designing a distributed algorithm to obtain the Euclidean gradients of precoder $\tilde{\mathbf{p}}_l$ for a given criterion J with a τ-length training pilot sequence $\mathbf{s}_k \in \mathbb{C}^{\tau\times 1}, (k = 1, \cdots, K)$:

1) Each UE adopts the MMSE combiner according to (6), which is implemented locally with the training pilot sequence.

2) Each UE transmits the beamformed gradient to the APs:

$$\mathbf{x}_k^{\mathrm{ul}}(i) = \frac{1}{\tau}\mathbf{w}_k(\hat{s}_k(i) - s_k(i))\frac{\partial J}{\partial e_k} \quad (i = 1, \cdots, \tau). \tag{25}$$

And AP l receives

$$\mathbf{y}_l^{\mathrm{ul}}(i) = \frac{1}{\tau}\sum_{k=1}^{K}\mathbf{H}_{k,l}^H\mathbf{w}_k(\hat{s}_k(i) - s_k(i))\frac{\partial J}{\partial e_k} + \mathbf{n}_l^{\mathrm{ul}}(i), \tag{26}$$

where $\mathbf{n}_l^{\mathrm{ul}}(i) \sim \mathcal{CN}(0, \sigma_l^2\mathbf{I})$ is the AWGN at the receiver of AP l. Since the first term in (26) after averaging over the τ-length pilot sequence is exactly the gradient $\frac{\partial J}{\partial \mathbf{x}_l^*}$ as given in (17). AP l can obtain the gradient $\frac{\partial J}{\partial \mathbf{x}_l^*}$ from $\mathbf{y}_l^{\mathrm{ul}}$, where the channel noise can also be largely mitigated when averaging over a τ-length pilot sequence to have

$$\nabla J(\tilde{\mathbf{p}}_l) = \frac{1}{\tau}\sum_{i=1}^{\tau}\frac{\partial J(i)}{\partial \tilde{\mathbf{p}}_l^*(i)} \approx \frac{1}{\tau}\sum_{i=1}^{\tau}(\mathbf{s}_l(i)^* \otimes \mathbf{A}^T)\mathbf{y}_l^{\mathrm{ul}}(i) \tag{27}$$

Because $\mathbf{s}_l(i), (i = 1, \cdots, \tau)$ is locally available, each AP can obtain the gradient $\frac{\partial J}{\partial \tilde{\mathbf{p}}_l^*}$ by (16), without knowing the CSI.

After obtaining the Euclidean gradient (27), AP l calculates the Riemannian gradient $\mathrm{grad}\, J(\tilde{\mathbf{p}}_l)$ by projecting it onto the tangent space $T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l$. For the functions that can be extended to the ambient Euclidean space, the relation between the Riemannian gradient and the Euclidean gradient is [26]

$$\mathrm{grad}\, J(\tilde{\mathbf{p}}_l) = \mathrm{Proj}_{T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l}(\nabla J(\tilde{\mathbf{p}}_l)). \tag{28}$$

Here, $\mathrm{Proj}_{T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l}(\cdot)$ denotes the orthogonal projection onto $T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l$, and is given by [27]

$$\mathrm{Proj}_{T_{\tilde{\mathbf{p}}_l}\mathcal{M}_l}(\boldsymbol{\psi}) = \boldsymbol{\psi} - \Re\left\{\tilde{\mathbf{p}}_l^H\boldsymbol{\psi}\right\}\tilde{\mathbf{p}}_l. \tag{29}$$

Therefore, the Riemannian gradient of objective J on manifold $\mathcal{M}_l$ at $\tilde{\mathbf{p}}_l$ is

$$\mathrm{grad}\, J(\tilde{\mathbf{p}}_l) = \nabla J(\tilde{\mathbf{p}}_l) - \Re\left\{\tilde{\mathbf{p}}_l^H\nabla J(\tilde{\mathbf{p}}_l)\right\}\tilde{\mathbf{p}}_l. \tag{30}$$

Subsequently, $\mathrm{grad}\, J(\tilde{\mathbf{P}}_l)$ and $\nabla J(\tilde{\mathbf{P}}_l)$ are obtained from reshaping $\mathrm{grad}\, J(\tilde{\mathbf{p}}_l)$ and $\nabla J(\tilde{\mathbf{p}}_l)$, respectively.

Now AP l can update its precoder with the first-order Riemannian gradient descent as:

$$\tilde{\mathbf{P}}_l(t) = \mathrm{Retr}\left[\tilde{\mathbf{P}}_l(t-1) - \alpha\, \mathrm{grad}\, J(\tilde{\mathbf{P}}_l(t))\right], \tag{31}$$

with $\mathrm{Retr}[\cdot]$ denoting the retraction operation

$$\mathrm{Retr}[\tilde{\mathbf{P}}_l] = \frac{\tilde{\mathbf{P}}_l}{\|\tilde{\mathbf{P}}_l\|_F}. \tag{32}$$

While the UEs can calculate their MMSE receivers according to (6) based on the pilot transmission in the downlink, the above discussions indicate that the APs can optimize their precoders based on the gradient transmission in the uplink, as shown in Fig. 3.

The overall procedure is summarized in Algorithm 1, which consists of T rounds of DL-UL ping-pong iterations conducted *over-the-air*. The DQNPM algorithm is highly similar to the routine of ANN training using the forward transmission of training data and the back-propagation of derivatives. This explains why we refer to the proposed method as a quasi-NN approach.

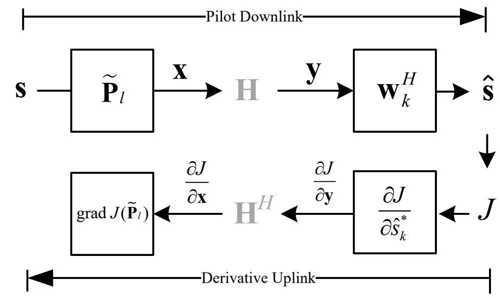


Fig. 3: Illustration of the Distributed Quasi-neural Network Precoding on Manifold Iteration

Algorithm 1 DQNPM Algorithm

Input: Pilot sequences $\mathbf{s}_1, \cdots, \mathbf{s}_K$ associated with K users

Output: Precoders $\mathbf{P}_l,\ (l = 1, \cdots, L)$

Initialize: Randomly initilize $\mathbf{P}_l$ to a feasible one $(l = 1, \cdots, L)$, $t = 0$;

For $t < T$ **do**:

1. $t \leftarrow t + 1$.
2. **DL**: Each AP transmits beamformed signal $\mathbf{x}_l$ by (13).
3. Each UE recovers $\hat{s}_k$ by applying the MMSE combiner $\mathbf{w}_k$ as (6).
4. **UL**: Each UE transmits the uplink signal $\mathbf{x}_k^{\mathrm{ul}}$ as (25).
5. Each AP computes gradient $\frac{\partial J}{\partial \tilde{\mathbf{p}}_l^*}$ by (27)
6. Each AP calculates the Riemannian gradient by (30) and reshapes to obtain $\mathrm{grad}\, J(\tilde{\mathbf{P}}_l)$.
7. Each AP updates $\tilde{\mathbf{P}}_l$ by (31).

End For

C. Computational Complexity, Overhead, and Scalability

As a distributed algorithm, the computational burden is apportioned among the nodes. On the UE side, the computational complexity is $\mathcal{O}(M_r^3)$ to compute the MMSE receiver $\mathbf{w}_k$ in (6) and $\mathcal{O}(M_r)$ to compute the beamformed gradient $\mathbf{x}_k^{\mathrm{ul}}$ in (25) for uplink transmission. Hence the overall complexity in each iteration is $\mathcal{O}(M_r^3)$ for each UE. Since the gradients is transmitted over-the-air, the summations and matrix-vector multiplications in (26) are achieved automatically. On the AP side, the optimization on manifold $\mathcal{M}_l$ only performs matrix-vector multiplications and requires the complexity of $\mathcal{O}(M_t N_l)$. On the other hand, its counterpart, the Distributed-OTA algorithm [8], requires matrix inverse both at the UEs and APs and has the computational complexity of $\mathcal{O}(M_r^3)$ and $\mathcal{O}(M_t^3)$ respectively.

For the OTA training overhead, each iteration of the DQNPM algorithm consists of one downlink transmission and one uplink transmission. While the Distributed-OTA method [8] requires one downlink transmission and two uplink transmissions. For the fronthaul overhead, both the DQNPM algorithm and the Distributed-OTA method [8] require no fronthaul signaling for CSI and data sharing.

The scalability of the CF-mMIMO networks requires finite computational complexity and resources in *i)*. signal processing for channel estimation; *ii)*. signal processing for data

This article has been accepted for publication in IEEE Transactions on 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/TCOMM.2024.3443735

reception and transmission; *iii)*. fronthaul signaling for data and CSI sharing; *iv)*. power allocation optimization. Since the DQNPM algorithm requires no explicit channel estimation nor fronthaul signaling for data and CSI sharing, the complexity and resource requirements for these purposes are negligible. Under the user-centric clustering scenario, the number of serving UE N_l is a finite number for each AP, and thus the computational complexity only scales with the number of transmitting/receiving antennas. Meanwhile, the DQNPM algorithm automatically allocates the transmitting power while optimizing the beamforming weights. Hence, the DQNPM algorithm is salable in the user-centric CF-mMIMO network.

IV. Extension Studies

In the previous section, the DQNPM algorithm is derived for the case where each UE is allocated only one stream. In this section, we first extend it to the multi-stream case, and then delineate a category of cost functions $J\left(e_1(\{\mathbf{P}_l\}_{l\in\mathcal{L}}),\cdots,e_K(\{\mathbf{P}_l\}_{l\in\mathcal{L}})\right)$ that allow for fully-distributed optimization. Finally, we explain that the DQNPM can be incorporated into a 5G-New Radio (NR) TDD system by piggybacking its frame structure.

A. Extension to Multi-stream per UE Case

Denote $\mathbf{d}_k \in \mathbb{C}^{M_s\times 1}$ as the data streams for UE k with $M_s \le M_r$. By a predefined allocation strategy, AP l processes signal $\mathbf{s}_l \in \mathbb{C}^{N_l M_s\times 1}$ from $\mathcal{D}=\{\mathbf{d}_1,\cdots,\mathbf{d}_K\}$ with precoding matrix $\mathbf{P}_l \in \mathbb{C}^{M_t\times N_l M_s}$. The combined signal at UE k is

$$\hat{\mathbf{d}}_k = \mathbf{W}_k^H\mathbf{y}_k = \mathbf{W}_k^H\left(\sum_{l=1}^{L}\mathbf{H}_{k,l}\mathbf{P}_l\mathbf{s}_l + \mathbf{n}_k\right) \in \mathbb{C}^{M_s\times 1}. \tag{33}$$

Here, the MMSE receiver is [cf. (6)]

$$\mathbf{W}_k = \left(\mathbf{Y}_k\mathbf{Y}_k^H\right)^{-1}\mathbf{Y}_k\mathbf{D}_k^H, \tag{34}$$

where $\mathbf{D}_k \in \mathbb{C}^{M_s\times\tau}$ is the downlink τ-length pilot and $\mathbf{Y}_k \in \mathbb{C}^{M_r\times\tau}$ is the received signal by UE k.

Let $\boldsymbol{\Psi}_k(i) \triangleq (\hat{\mathbf{d}}_k(i)-\mathbf{d}_k(i))(\hat{\mathbf{d}}_k(i)-\mathbf{d}_k(i))^H$, $i=1,\cdots,\tau$ and the MSE matrix $\mathbf{E}_k = \frac{1}{\tau}\sum_{i=1}^{\tau}\boldsymbol{\Psi}_k(i)$. The precoding problem in the multi-stream per UE case is [cf. (15)]

$$\begin{aligned}\min_{\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}}\quad & J(\mathbf{E}_1(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}),\cdots,\mathbf{E}_K(\{\tilde{\mathbf{p}}_l\}_{l\in\mathcal{L}}))\\ \text{s.t.}\quad & \tilde{\mathbf{p}}_l \in \mathcal{M}_l,\quad (l=1,\cdots,L).\end{aligned} \tag{35}$$

According to the chain rule, the gradient with respect to the transmitting signal $\mathbf{x}_l$ is [cf. (17)]

$$\frac{\partial J}{\partial \mathbf{x}_l^*(i)} = \frac{1}{\tau}\sum_{k=1}^{K}\mathbf{H}_{k,l}^H\mathbf{W}_k\frac{\partial J}{\partial \mathbf{E}_k}(\hat{\mathbf{d}}_k(i)-\mathbf{d}_k(i)). \tag{36}$$

For the MWSR criterion, the cost function is defined as [cf. (8)]:

$$J \triangleq \sum_{k=1}^{K}\omega_k\log\det\left(\mathbf{E}_k\right). \tag{37}$$

And the gradient with respect to $\mathbf{E}_k$ is [cf. (23)]

$$\frac{\partial J}{\partial \mathbf{E}_k} = \omega_k\mathbf{E}_k^{-1}. \tag{38}$$

For the MPF criterion, the cost function is defined as [cf. (10)]:

$$J \triangleq -\sum_{k=1}^{K}\log\left(-\log\det\left(\mathbf{E}_k\right)\right). \tag{39}$$

And the derivative is [cf. (24)]

$$\frac{\partial J}{\partial \mathbf{E}_k} = -\frac{1}{\log\det(\mathbf{E}_k)}\mathbf{E}_k^{-1}. \tag{40}$$

The DQNPM algorithm in the multi-stream scenario is executed similarly as that in the single stream per UE case and operates with T iterations, each with a τ-length pilot sequence. AP l transmits the beamformed signal $\mathbf{A}\tilde{\mathbf{P}}_l\mathbf{s}_l$ as (13); UE k applies the combiner $\mathbf{W}_k$ as (34) to recovers $\hat{\mathbf{d}}_k$ as (33). In the uplink transmission, each UE transmits the beamformed gradient $\frac{1}{\tau}\mathbf{W}_k\frac{\partial J}{\partial \mathbf{E}_k}(\mathbf{d}_k(i)-\hat{\mathbf{d}}_k(i))$. Each AP executes exactly the same procedures as in the single-stream case and updates its local beamforming weight distributively.

B. Cost Functions that Allow for Distributed Optimization

Besides the MWSR and the MPF criteria, the following proposition delineates a large category of cost functions $J(e_1(\{\mathbf{P}_l\}_{l\in\mathcal{L}}),\cdots,e_K(\{\mathbf{P}_l\}_{l\in\mathcal{L}}))$ that can be accommodated by the DQNPM (and the DQNP algorithm alike) for distributed optimization of the CF-mMIMO network.

Proposition 2. *The cost function*

$$J(e_1(\{\mathbf{P}_l\}_{l\in\mathcal{L}}),\cdots,e_K(\{\mathbf{P}_l\}_{l\in\mathcal{L}}))$$

can be distributively minimized, if

i) it can be factored into

$$u\left(g_1(e_1),\ldots,g_K(e_K)\right), \tag{41}$$

where $g_k(\cdot):\mathbb{R}\mapsto\mathbb{R}$ is a differentiable function that only needs to be known locally to UE k, and

ii) $u(\cdot):\mathbb{R}^K\mapsto\mathbb{R}$ is differentiable with its partial differentials satisfying

$$\frac{\partial u(g_1,\ldots g_K)}{\partial g_k} = \frac{\partial u(g_1,\ldots g_K)}{\partial g_{k'}} = d(\{e_k\}_{k\in\mathcal{K}}) > 0 \tag{42}$$

for any $k'\neq k$, where $d(\{e_k\}_{k\in\mathcal{K}})$ is a positive constant.

Proof. Similar to the proof of Proposition 1, we consider the case $e_k = |\hat{s}_k - s_k|^2$, and the case $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2]$ can be similarly extended. Note that

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{P}_l^*} &= \sum_{k=1}^{K}\frac{\partial J}{\partial e_k}\cdot\frac{\partial e_k}{\partial \mathbf{P}_l^*} = \sum_{k=1}^{K}\frac{\partial J}{\partial u}\frac{\partial u}{\partial g_k}\frac{\partial g_k}{\partial e_k}\cdot\frac{\partial e_k}{\partial \mathbf{P}_l^*}\\ &= d(\{e_k\}_{k\in\mathcal{K}})\sum_{k=1}^{K}\mathbf{H}_{k,l}^H\mathbf{w}_k\frac{\partial g_k}{\partial e_k}(\hat{s}_k - s_k)\mathbf{s}_l^H\end{aligned} \tag{43}$$

where the last equality follows from (42) and (1)–(3).

Since $g_k(e_k)$ is locally known to the UE k, so is $\frac{\partial g_k(e_k)}{\partial e_k}$. Therefore, if every of the UE sends uplink signaling $\mathbf{w}_k\frac{\partial g_k}{\partial e_k}(\hat{s}_k - s_k)$ simultaneously, then the AP l receives

$$\sum_{k=1}^{K}\mathbf{H}_{k,l}^H\mathbf{w}_k\frac{\partial g_k}{\partial e_k}(\hat{s}_k - s_k). \tag{44}$$

This article has been accepted for publication in IEEE Transactions on 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/TCOMM.2024.3443735

After multiplying the local source stream $\mathbf{s}_l^H$ to the right, it is proportional to $\frac{\partial J}{\partial \mathbf{P}_l^*}$. Therefore, the APs can update their precoding matrices using a gradient descent method in a distributed manner.

□

About the MWSR criterion (8), letting $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2]$. The cost function $J = \sum_{k=1}^K \omega_k \log(e_k)$ can be factorized into $u(g_1(e_1), \ldots, g_K(e_K)) = \sum_{k=1}^K g_k(e_k)$ with $g_k(e_k) = \omega_k \log(e_k)$. It is obvious that $\frac{\partial u(g_1, \ldots g_K)}{\partial g_k} = 1$ for any k. Hence, both conditions of Proposition 2 are satisfied.

About the MPF criterion (10), letting $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2]$, the cost function $J = -\sum_{k=1}^K \log(-\log(e_k))$ can be factorized into $u(g_1(e_1), \ldots, g_K(e_K)) = \sum_{k=1}^K g_k(e_k)$ with $g_k(e_k) = -\log(-\log(e_k))$. It is obvious that $\frac{\partial u(g_1, \ldots g_K)}{\partial g_k} = 1$ for any k. Hence, the conditions of Proposition 2 are satisfied.

Proposition 2 delineates a broad category of cost functions. Besids the MWSR and the MPF criterion, we can easily show that the following examples.

1) MMSE problem: The MMSE problem is an obvious example. Specifically, choose $e_k = |\hat{s}_k - s_k|^2$ and let $u(g_1(e_1), \ldots, g_K(e_K)) = \sum_{k=1}^K g_k(e_k)$, $g_k(e_k) = \omega_k e_k$, the MMSE problem is given as

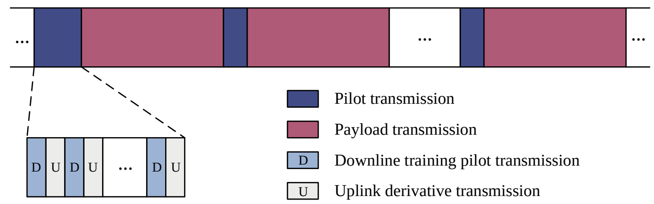
$$\min J = \sum_{k=1}^{K} \omega_k e_k. \tag{45}$$

2) Maximum harmonic rate: The maximum harmonic rate (MHR) problem is given as $\max K(\sum_{k=1}^K \mathrm{SE}_k^{-1})^{-1}$. Choose $e_k = \mathbb{E}[|\hat{s}_k - s_k|^2]$ and let $u(g_1(e_1), \ldots, g_K(e_K)) = -\frac{K}{\sum_{k=1}^K g_k(e_k)}$ with $g_k(e_k) = -\frac{1}{\log(e_k)}$, the MHR problem can be reformulated as

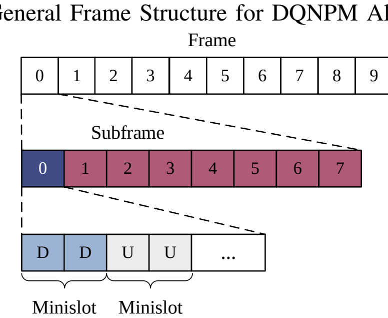
$$\min J = K \left(\sum_{k=1}^{K} \frac{1}{\log(e_k)} \right)^{-1}. \tag{46}$$

C. Practical Implementation with Frame Structure

As the DQNPM algorithm exploits the channel reciprocity of a TDD system, we can implement it in a practical 5G-NR TDD system by designing the frame structure as illustrated in Fig. 4, which is similar to [8, Sec. 5] and [21, Sec. 3]. As shown in Fig. 4b, a 5G-NR frame is made up of 10 subframes, each consisting of 8 time slots, where the first is for the pilot and the subsequent seven are for the data transmission. Furthermore, a slot spans 14 orthogonal frequency division multiplexing (OFDM) symbols, and every minislot, which consists of two OFDM symbols, can be used for either an uplink training or a downlink training. To be more specific, a downlink minislot is used to transmit the beamformed signal, and an uplink minislot is used to transmit the beamformed gradients – cf. Line 2 and Line 4 in Algorithm 1, respectively. Therefore, one subframe can support up to $7/2 * 8 = 28$ rounds of iterations. In practice the switching time between the downlink and the uplink signaling will consume some time resource; thus, the number of iterations in each subframe will be a smaller number. Due to the time-varying property of wireless channels, the system needs to be retrained periodically to maintain a high performance. But the subsequent training typically requires fewer training iterations than the initial one since the previous optimized weights can be considered as a "warm start".



(a) General Frame Structure for DQNPM Algorithm



(b) 5G NR Frame Structure for DQNPM Algorithm

Fig. 4: Frame Structure Design for DQNPM Algorithm

V. Simulation Results

This section verifies the performance of our proposed algorithm by numerical simulations.

Let $\mathcal{L}_k$ denote the set of APs serving UE k. The rate of UE k can be given by [28]

$$R_k = \log_2 \left| \mathbf{I} + \mathbf{W}_k \mathbf{C}_k^{-1} \mathbf{W}_k^H \hat{\mathbf{H}}_{k,k} \hat{\mathbf{P}}_k \hat{\mathbf{P}}_k^H \hat{\mathbf{H}}_{k,k}^H \right|, \tag{47}$$

where the covariance matrix of interference-plus-noise

$$\mathbf{C}_k = \sum_{\bar{k} \neq k} \mathbf{W}_k^H \hat{\mathbf{H}}_{k,\bar{k}} \hat{\mathbf{P}}_{\bar{k}} \hat{\mathbf{P}}_{\bar{k}}^H \hat{\mathbf{H}}_{k,\bar{k}}^H \mathbf{W}_k + \sigma_k^2 \mathbf{W}_k^H \mathbf{W}_k. \tag{48}$$

Here, $\hat{\mathbf{H}}_{k,\bar{k}} = [\mathbf{H}_{k,l}]_{l \in \mathcal{L}_{\bar{k}}} \in \mathbb{C}^{M_r \times |\mathcal{L}_{\bar{k}}| M_t}$ is the aggregated channel matrix of the UE $\bar{k}$. $\hat{\mathbf{P}}_{\bar{k}} = \left[\mathbf{P}_{\bar{k},l}^H \right]_{l \in \mathcal{L}_{\bar{k}}}^H \in \mathbb{C}^{|\mathcal{L}_{\bar{k}}| M_t \times M_s}$ is the aggregated precoding matrix for UE $\bar{k}$. Hence, the weighted sum-rate is

$$R = \sum_{k \in \mathcal{K}} \omega_k R_k. \tag{49}$$

The simulated CF-mMIMO network consists of $L = 100$ APs each equipped with $M_t = 4$ (unless otherwise specified) antennas located in a square grid with inter-site distance 100 m. The APs serve $K = 50$ UEs randomly located in the area. The UEs are equipped with $M_r = 2$ (unless otherwise specified) antennas. The heights of each AP and UE are 10 m and 1 m, respectively. We consider the Rayleigh fading channel model with carrier frequency 3.6 GHz that $\mathrm{vec}(\mathbf{H}_{k,l}) \sim \mathcal{CN}(0, \delta_{k,l} \mathbf{I}_{M_r M_t})$, where $\delta_{k,l}[\mathrm{dB}] = -30.5 - 36.7 \log_{10}(r_{k,l})$ and $r_{k,l}$ is the distance between UE k and AP l. Each AP uses downlink transmit power 30 dBm, each UE

This article has been accepted for publication in IEEE Transactions on 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/TCOMM.2024.3443735

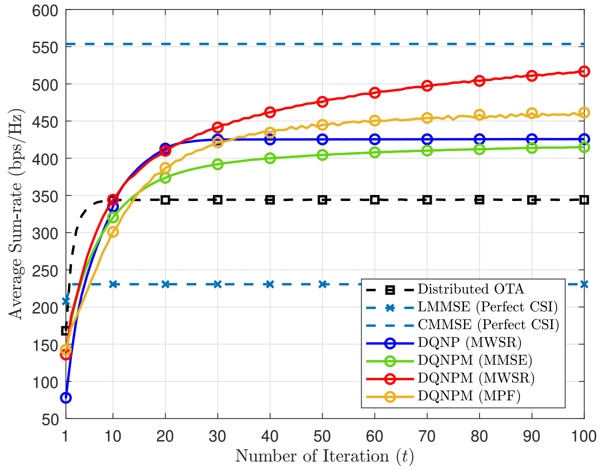


Fig. 5: Convergence of average sum-rate for DQNPM and DQNP algorithm

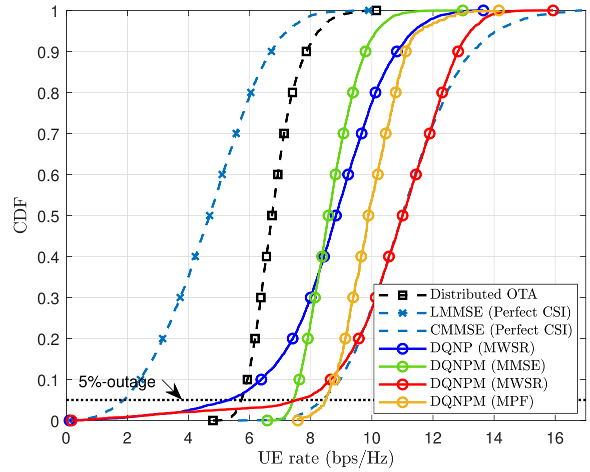
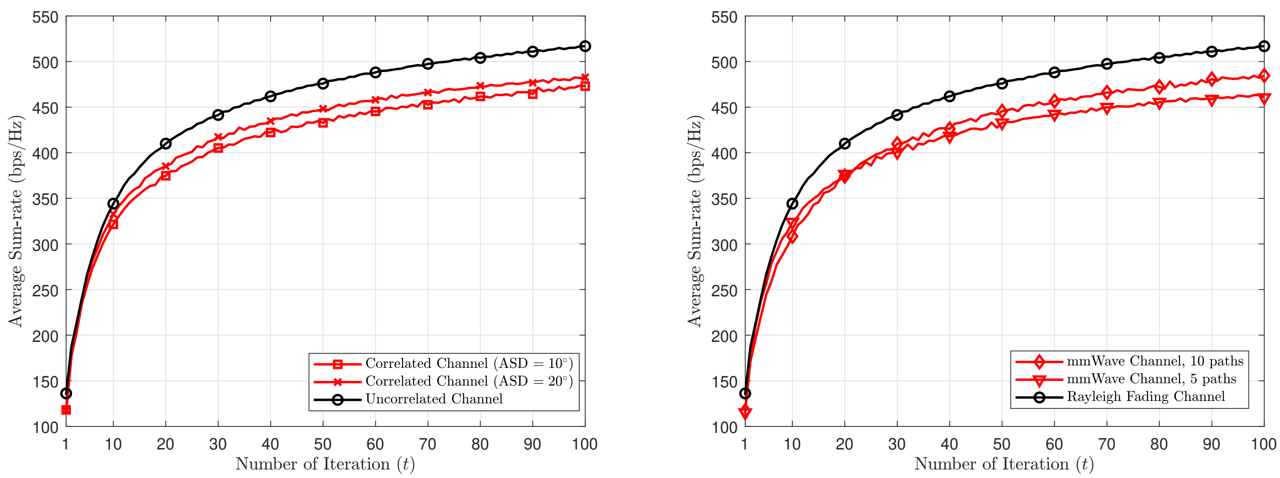


Fig. 6: CDF of per-UE rate for DQNPM and DQNP algorithm



(a) Performance with correlated Rayleigh fading channel

(b) Performance with mmWave multipath channel

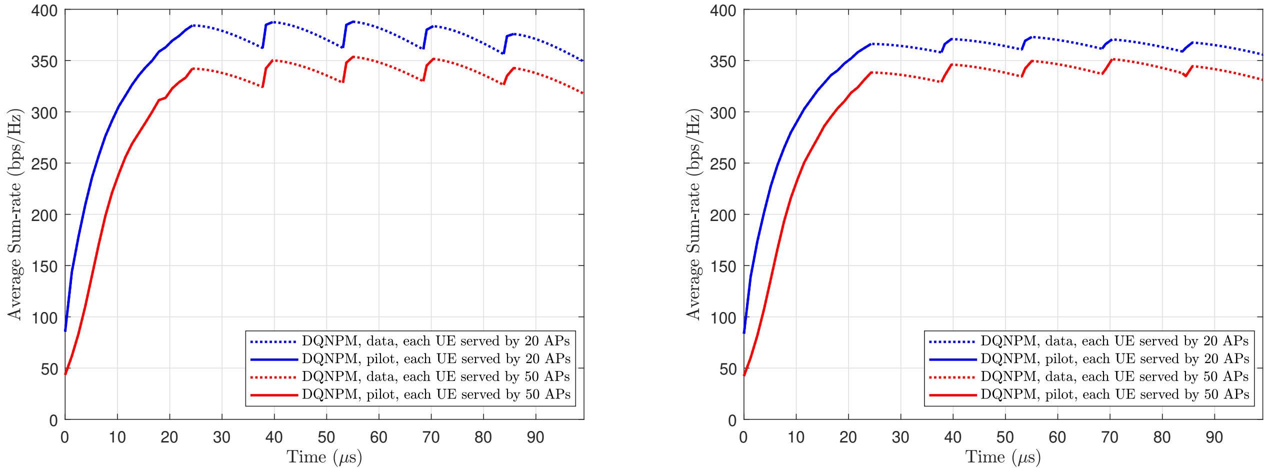
Fig. 7: DQNPM algorithm's performance with other channel models

uses uplink transmit power 20 dBm, and the noise power is set as $\sigma_l^2 = \sigma_k^2 = -95$ dBm.

We use 100 Monte Carlo simulations with different channel realizations and UE drops to evaluate the system performance. For each AP, the closest 20 UEs are chosen to be served (unless otherwise stated), and the length of pilots $\tau = 128$ (unless otherwise stated).

In the first simulation, we evaluate the performance of the DQNPM algorithm with the MWSR criterion in (8) and the MPF criterion in (10). As shown in Fig. 5, the DQNPM algorithm with the MWSR criterion outperforms that with the MMSE criterion. It's also interesting to observe that the DQNPM algorithm outperforms the DQNP algorithm. The DQNPM algorithm significantly outperforms the state-of-the-art approaches, i.e., the Distributed-OTA algorithm [8] and the LMMSE algorithm [7], by 50.2% and 124.1% after convergence, and by 19.2% and 77.8% after 20 iterations. It can be seen that the DQNPM requires more iterations to achieve better performance at the initial stage. Meanwhile, it achieves a high performance close to the centralized minimum mean-square error (CMMSE) method since no inter-AP cooperation information is ignored. Fig. 6 shows the cumulative distribution function (CDF) of the rates of the UEs as given in (47). For a fair comparison with the Distributed-OTA algorithm [8], which is designed based on the MMSE criterion, the DQNPM algorithm with the MMSE criterion shows uniformly superior performance, with a 31.5% increase of the last 5% UE rate. It is interesting to observe that the DQNPM (and the DQNP algorithm as well) left out a small portion of the UEs (as their left tails reach close to zero) for sum-rate maximization. This problem can be avoided by choosing the MPF objective function (10), which achieves the highest last 5% UE rate among the simulated algorithms, which is 48.6% higher than that achieved by the Distributed-OTA algorithm [8].

To further verify the performance of the DQNPM algorithm in different communication environments, we simulated



(a) $f_c = 3.6$ GHz, $v = 90$ m/s

(b) $f_c = 30$ GHz, $v = 10$ m/s

Fig. 8: DQNPM algorithm's performance with time-varying channel

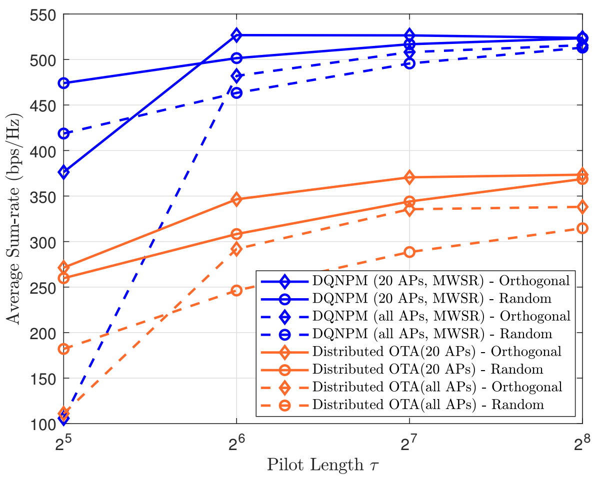


Fig. 9: Average sum-rate versus training pilot length τ

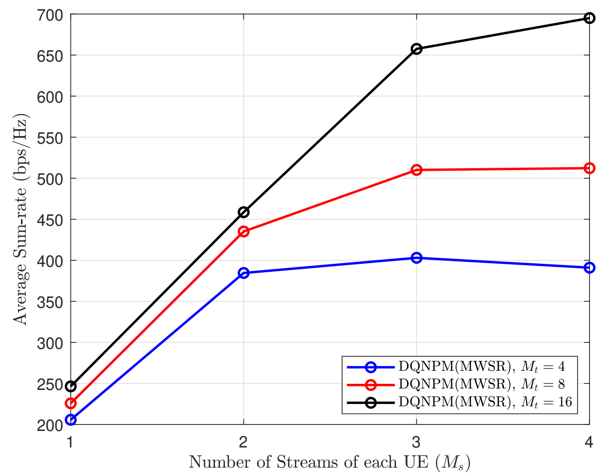


Fig. 10: Average sum-rate versus number of UE streams M_s

the DQNPM algorithm with the correlated Rayleigh fading channel $\text{vec}(\mathbf{H}_{k,l}) \sim \mathcal{CN}(0, \mathbf{R}_{k,l})$, where $\mathbf{R}_{k,l}$ describes channel spatial correlation. Specifically, we used the local scattering model [28] for the spatial correlation matrix with the different angular standard deviation (ASD). Fig. 7(a) shows that the DQNPM algorithm applies to both the correlated and uncorrelated channel models. However, due to the effect of spatial-correlated channels, a lower average sum-rate is achieved. In addition, we simulated the DQNPM algorithm with millimeter wave (mmWave) channels [29] with carrier frequency 30 GHz as shown in Fig. 7(b). These results verify the wide range of applicability of the DQNPM algorithm.

Furthermore, the proposed algorithm is evaluated in a more practical scenario, the time-varying channel. Let the carrier frequency $f_c = 3.6$ GHz and $f_c = 30$ GHz. Let the signal bandwidth $B = 100$ MHz with the Nyquist sampling duration $T_s = \frac{1}{B} = 0.01$ µs. To evaluate the performance of time-varying channels with different carrier frequencies, we consider Clarke's model [30] for the time-varying Rayleigh fading channel with the 3.6 GHz band; and the time-varying mmWave channel [31] for the 30 GHz band. To realize a similar time-varying effect, we set the channel mobility speed $v = 10$ m/s for carrier frequency 30 GHz and $v = 90$ m/s for carrier 3.6 GHz. In both cases, the corresponding Doppler frequency spread $f_d = \frac{v}{c} f_c \approx 1$ kHz, where c is the speed of the magnetic wave. We use 20 sets of pilots with length $\tau = 64$ for the initial training, which lasts $20 \times 64 \times 2 \times 0.01 = 25.6$ µs. After that, 20 sets of data are transmitted, which lasts for $20 \times 64 \times 0.01 = 12.8$ µs. As shown in Fig. 8, due to the time-varying channel, the performance of the downlink transmission is slowly degrading with less than 5% until the subsequent training. Based on the precoding weights obtained by the initial training, the "warm-start" training only takes two iterations to achieve the initial performance and is followed by another 20 sets of downlink data. This simulation shows that the proposed DQNPM algorithm only requires a moderate amount of resources to adjust to the time-varying channel in the CF-mMIMO network. Meanwhile, serving a proper number of UEs facilitates the initial training process. Hence, the case that each UE is served with the 20 closest APs outperforms that with the 50 closest APs.

We then consider the scenario with pilot contamination by implementing non-orthogonal random pilots. We consider both non-orthogonal random pilots and orthogonal pilots with varying length τ as shown in Fig. 9. We also consider two serving scenarios, i.e. each UE served by the closest 20 APs (solid curves) and each UE served by all 100 APs (dashed curves). First, we consider the impact of different pilot length. For $\tau = 32$, using orthogonal pilots will suffer from extensive pilot contamination because the same orthogonal pilots must be reused, which explains the reason for the severe decrease for both the DQNPM and the distributed-OTA [8] algorithm as τ reduce from 64 to 32. Meanwhile, compared with the scenario of each UE served by the closest 20 APs, the scenario of each UE served by all APs has higher density and introduces more pilot contamination. For $\tau \geq 64$, the performance gap between using random pilots and using orthogonal pilots of the DQNPM algorithm is closer than that of the distributed-OTA algorithm [8]. This indicates that the DQNPM algorithm is more robust to the pilot contamination caused by the non-orthogonal pilots. This simulation also shows that choosing a proper AP-UE pairing is essential to balance between the impact of unnecessary connections and the gain of coordinated communication, especially when the number of orthogonal pilots is less than that of the active UEs.

We finally evaluate the performance of the DQNPM algorithm when each UE is allocated with multiple data streams. Here the network consists of $L = 25$ APs equipped with M_t transmitting antennas ($M_t = 4, 8, 16$) and $K = 20$ UEs equipped with $M_r = 4$ receiving antennas. M_s independent streams ($M_s = 1, 2, 3, 4$) are transmitted to each UE, and each UE is served by 25 APs. Fig. 10 shows the average sum rate of the CF-mMIMO network by the DQNPM algorithm with the MWSR criterion after convergence. As M_t increases, the CF-mMIMO network can support more data streams and achieve a higher average sum rate.

VI. Conclusion and Discussion

This paper investigates the distributed precoding problem in a downlink CF-mMIMO network and proposes a distributed Quasi-NN precoding algorithm, which borrows the idea of the BP algorithm. Using OTA training pilots transmission to optimize the precoding weights, it requires no explicit channel estimation nor data sharing via fronthaul links. The proposed algorithm can be applied based on the 5G NR frame structure and accommodates a large variety of objective functions. Numerical simulations show the effectiveness of the proposed schemes and their superiority over state-of-art methods.

The proposed algorithm relates a CF-mMIMO network to an ANN; thus, we refer to it as a Quasi-NN approach. But it is a model-based approach and is fully-interpretable. The key of this approach is to model a signal processing problem or a system optimization problem by a layered structure like an ANN, which indeed indicates a new road to solve other challenging signal processing problems, such as spectral estimation [22] and optimal design of a highly-nonlinear receiver [32].

Appendix A. About Complex Matrix Gradient

For complex matrix gradient, it has the following definition and lemma [33]:

Definition 1 (Gradient of complex variables [33])**.** *Given $z = a + jb$, where $a, b \in \mathbb{R}$, the gradients with respect to z and z^* of $f(z_0)$ at $z_0 \in \mathbb{C}$ are defined as*

$$\frac{\partial f(z_0)}{\partial z} = \frac{1}{2}\left(\frac{\partial f(z_0)}{\partial a} - j\frac{\partial f(z_0)}{\partial b}\right), \tag{50}$$

and

$$\frac{\partial f(z_0)}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f(z_0)}{\partial a} + j\frac{\partial f(z_0)}{\partial b}\right). \tag{51}$$

Here, the variables z and z^* are considered independent.

Lemma 1 ([33], Theorem 3.3)**.** *For $f : \mathbb{C}^{N\times Q} \times \mathbb{C}^{N\times Q} \to \mathbb{R}$, we have*

$$\frac{\partial f}{\partial \mathbf{Z}^*} = \left(\frac{\partial f}{\partial \mathbf{Z}}\right)^*. \tag{52}$$

Appendix B. The DQNP Algorithm

In this appendix, we introduce the DQNP algorithm. Since the precoder $\mathbf{P}_l$ consists of both the amplitude and the direction precoding, it can be further decomposed into two parts as

$$\mathbf{P}_l = e^{-|\eta_l|}\frac{\mathbf{V}_l}{\|\mathbf{V}_l\|_F}. \tag{53}$$

Here, $\mathbf{V}_l$ is intended to control the direction of the transmitted signal and η_l is a power control factor for AP l with $e^{-|\eta_l|} \in [0, 1]$, which follows the power constraint.

Hence, the constrained optimization problem (11) can be reformulated into an unconstrained one

$$\min_{\{\mathbf{V}_l,\eta_l\}_{l\in\mathcal{L}},\{\mathbf{w}_k\}_{k\in\mathcal{K}}} f(\mathbf{V}_l, \eta_l, \mathbf{w}_k). \tag{54}$$

Although the precoding design problem can be modeled as an unconstrained problem with three optimization parameters, a distributed solution for problem (11) is still challenging due to the limited CSI at each AP. Similar to the DQNPM algorithm, we can propose the distributed pilot-aided algorithm for optimizing the CF-mMIMO network under the Quasi-NN and have the following gradient results.

Proposition 3. *The gradient with respect to the combiner $\mathbf{w}_k$ of UE k is:*

$$\frac{\partial J}{\partial \mathbf{w}_k^*} = \mathbf{y}_k\left(\frac{\partial J}{\partial \hat{s}_k^*}\right)^*. \tag{55}$$

The gradient with respect to the direction precoder $\mathbf{V}_l$ of AP l is given for each row that

$$\frac{\partial J}{\partial \mathbf{v}_{ln}^H} = \frac{\partial J}{\partial x_{ln}^*}\frac{\partial x_{ln}^*}{\partial \mathbf{v}_{ln}^H} + \frac{\partial J}{\partial x_{ln}}\frac{\partial x_{ln}}{\partial \mathbf{v}_{ln}^H}. \tag{56}$$

Here, $\mathbf{v}_{ln}$ is the n-th column of $\mathbf{V}_l^H$ with $\mathbf{V}_l = [\mathbf{v}_{l1}^H; \cdots ; \mathbf{v}_{lM_t}^H]^H$ and x_{ln} is the n-th element of transmitted signal $\mathbf{x}_l$. Meanwhile, $\frac{\partial J}{\partial \mathbf{x}_l^*}$ is given by eq. (17). And

$$\frac{\partial x_{ln}}{\partial \mathbf{v}_{ln}^H} = e^{-|\eta_l|} \frac{-\mathbf{v}_{ln}^T \mathbf{s}_l \operatorname{tr}\left(\mathbf{V}_l \mathbf{V}_l^H\right)^{-\frac{1}{2}} \mathbf{v}_{ln}^T}{2\|\mathbf{V}_l\|_F^2}, \quad (57)$$

$$\frac{\partial x_{ln}^*}{\partial \mathbf{v}_{ln}^H} = e^{-|\eta_l|} \frac{2\mathbf{s}_l^H \|\mathbf{V}_l\|_F - \mathbf{v}_{ln}^H \mathbf{s}_l^* \operatorname{tr}\left(\mathbf{V}_l \mathbf{V}_l^H\right)^{-\frac{1}{2}} \mathbf{v}_{ln}^T}{2\|\mathbf{V}_l\|_F^2}. \quad (58)$$

The gradient with respect to the power factor η_l of AP l is

$$\frac{\partial J}{\partial \eta_l} = -2\Re\left\{\operatorname{sign}(\eta_l) e^{-|\eta_l|} \left(\frac{\partial J}{\partial x_l^*}\right)^H \frac{\mathbf{V}_l}{\|\mathbf{V}_l\|_F} \mathbf{s}_l\right\}, \quad (59)$$

with $\operatorname{sign}(\cdot)$ denoting the sign function that outputs -1 if the input is negetive, and outputs 1 if the input is positive.

Proof. From (3) and Definition 1 in Appendix A, we have

$$\frac{\partial \hat{s}_k}{\partial \mathbf{w}_k^*} = \mathbf{y}_k, \quad \frac{\partial \hat{s}_k^*}{\partial \mathbf{w}_k^*} = 0. \quad (60)$$

By the chain rule, we obtain (55) immediately.

Based on Proposition 1, since $x_{ln} = e^{-|\eta_l|} \mathbf{v}_{ln}^T \mathbf{s}_l, n = 1, \cdots, M_t$ and $\|\mathbf{V}_l\|_F^2 = \operatorname{tr}\left(\mathbf{V}_l \mathbf{V}_l^H\right)$, we can prove gradients (57) and (58) by the quotient rule.

Now we prove the gradient with respect to the power factor η_l. By (1) and (53), we have gradient

$$\frac{\partial \mathbf{x}_l}{\partial \eta_l} = -\operatorname{sign}(\eta_l) e^{-|\eta_l|} \frac{\mathbf{V}_l}{\|\mathbf{V}_l\|_F} \mathbf{s}_l. \quad (61)$$

By chain rule and the conjugate property $\frac{\partial J}{\partial \mathbf{x}_l^T} \cdot \frac{\partial \mathbf{x}_l}{\partial \eta_l} = \left(\frac{\partial J}{\partial \mathbf{x}_l^T} \cdot \frac{\partial \mathbf{x}_l}{\partial \eta_l}\right)^*$, we can prove (59). □

Similar to the DQNPM algorithm, the DQNP algorithm is executed by *over-the-air* signal. Each training iteration requires a downlink signaling resource to propagate the pilots forward, and an uplink signaling resource to transmit gradients backward. By the end of each downlink-uplink propagation, each AP updates precoding weights $\mathbf{V}_l$ and η_l, and each UE updates combining weights $\mathbf{w}_k$ using the momentum gradient descent method [34] without estimation of the CSI. Meanwhile, the DQNP algorithm can be similarly embedded into the 5G-NR frame structure as illustrated in Sec. IV-C.

References

[1] W. Dai, J. Liu, R. Wang, and Y. Jiang, "Learning by Over-the-Air Training: Distributed Precoding for Cell-Free Massive MIMO," in *Proc. IEEE 24th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2023, pp. 76–80.

[2] M. Matthaiou, O. Yurduseven, H. Q. Ngo, D. Morales-Jimenez, S. L. Cotton, and V. F. Fusco, "The road to 6G: Ten physical layer challenges for communications engineers," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 64–69, 2021.

[3] Ö. T. Demir, E. Björnson, L. Sanguinetti *et al.*, *Foundations of user-centric cell-free massive MIMO*. Now Publishers, Inc., 2021.

[4] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO has unlimited capacity," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 574–590, 2017.

[5] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, "Ubiquitous cell-free massive MIMO communications," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, pp. 1–13, 2019.

[6] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G cellular networks: Resource allocation and comparison with the cell-free massive MIMO approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1250–1264, 2019.

[7] G. Interdonato, M. Karlsson, E. Björnson, and E. G. Larsson, "Local partial zero-forcing precoding for cell-free massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4758–4774, 2020.

[8] I. Atzeni, B. Gouda, and A. Tölli, "Distributed Precoding Design via Over-the-Air Signaling for Cell-Free Massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1201–1216, 2021.

[9] L. Du, L. Li, H. Q. Ngo, T. C. Mai, and M. Matthaiou, "Cell-free massive MIMO: Joint maximum-ratio and zero-forcing precoder with power control," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3741–3756, 2021.

[10] H. Liu, J. Zhang, S. Jin, and B. Ai, "Graph coloring based pilot assignment for cell-free massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9180–9184, 2020.

[11] S. Buzzi, C. D'Andrea, M. Fresia, Y.-P. Zhang, and S. Feng, "Pilot assignment in cell-free massive MIMO based on the Hungarian algorithm," *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 34–37, 2020.

[12] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, "Precoding and power optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4445–4459, 2017.

[13] S. Buzzi, C. D'Andrea, and C. D'Elia, "User-centric cell-free massive MIMO with interference cancellation and local ZF downlink precoding," in *15th Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*. IEEE, 2018, pp. 1–5.

[14] P. Liu, K. Luo, D. Chen, and T. Jiang, "Spectral efficiency analysis of cell-free massive MIMO systems with zero-forcing detector," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 795–807, 2019.

[15] E. Björnson and L. Sanguinetti, "Making cell-free massive MIMO competitive with MMSE processing and centralized implementation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 77–90, 2019.

[16] Ö. T. Demir, E. Björnson, L. Sanguinetti *et al.*, "Foundations of user-centric cell-free massive MIMO," *Found. and Trends® in Signal Process.*, vol. 14, no. 3-4, pp. 162–472, 2021.

[17] W. Ejaz, S. K. Sharma, S. Saadat, M. Naeem, A. Anpalagan, and N. A. Chughtai, "A comprehensive survey on resource allocation for CRAN in 5G and beyond networks," *J. Network Comput. Appl.*, vol. 160, p. 102638, 2020.

[18] C. Tao and S. Hamalainen, "Handover in IP Ran," in *Int. Conf. on Commun. Tech. Proc. (ICCT)*, vol. 2. IEEE, 2003, pp. 812–815.

[19] E. Björnson and L. Sanguinetti, "A new look at cell-free massive MIMO: Making it practical with dynamic cooperation," in *in Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*. IEEE, 2019, pp. 1–6.

[20] R. Wang, Y. Jiang, and W. Zhang, "Distributed Learning for MIMO Relay Networks," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 343–357, 2022.

[21] R. Wang, W. Dai, and Y. Jiang, "Distributed Learning for Uplink Cell-Free Massive MIMO Networks," *IEEE Trans. Commun.*, vol. 71, no. 9, pp. 5595–5606, 2023.

[22] Y. Jiang, T. Zhang, and W. Zhang, "Line Spectral Estimation Inspired by Quasi-Neural Network," *IEEE Trans. Signal Process.*, vol. 70, pp. 5822–5832, 2022.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[24] D. Guo, S. Shamai, and S. Verdu, "Mutual information and minimum mean-square error in Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1261–1282, 2005.

[25] J. M. Lee, *Introduction to Riemannian manifolds*. Springer, 2018, vol. 176.

[26] J. Hu, X. Liu, Z.-W. Wen, and Y.-X. Yuan, "A brief introduction to manifold optimization," *J. Oper. Res. Soc. China*, vol. 8, no. 2, pp. 199–248, 2020.

[27] C. Wang, A.-A. Lu, X. Gao, and Z. Ding, "Robust Precoding for 3D Massive MIMO Configuration With Matrix Manifold Optimization," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3423–3437, 2021.

[28] T. L. Marzetta and H. Yang, *Fundamentals of massive MIMO*. Cambridge University Press, 2016.

[29] O. El Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, "Spatially sparse precoding in millimeter wave mimo systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, 2014.

[30] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Novel sum-of-sinusoids simulation models for rayleigh and rician fading channels," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3667–3679, 2006.

[31] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.

[32] C. Wang, Q. Du, and Y. Jiang, "Optimal sequence detection for a one-bit adc receiver using a quasi-neural network," *IEEE Wireless Commun. Lett.*, vol. early access, 2024.

[33] A. Hjørungnes, *Complex-valued matrix derivatives: with applications in signal processing and communications*. Cambridge University Press, 2011.

[34] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Math. Program.*, vol. 156, no. 1-2, pp. 59–99, 2016.