无蜂窝大规模 MIMO 网络中的分布式功率控制学习

DaesungYu，IEEE 学生会员，Hoon Lee，IEEE 会员，
Seung-Eun Hong，以及 Seok-Hwan Park，IEEE 会员

arXiv:2303.02573v1 [eess.SP] 5 Mar 2023

摘要

本文研究了基于学习的无蜂窝大规模多输入多输出（MIMO）系统分布式功率控制方法。在这类系统中，中央处理器（CP）通过前传协调来控制接入点（AP）。为了确定分布式接入点的传输策略，必须建立一种全网协同优化机制。为应对这一挑战，我们设计了一种协作学习（CL）框架，通过专用的深度神经网络（DNN）模块管理 CP 和接入点的计算与协调策略。为了构建灵活的学习结构，我们对所提出的 CL 进行了精心设计，使其前向传播计算不受接入点数量的影响。为此，我们采用参数复用的概念，在所有接入点部署相同的 DNN 模块。因此，在特定配置下训练的所提 CL 可以直接应用于任意数量的接入点。数值结果验证了所提出的 CL 相较于传统非协作方法的优势。

本研究部分由韩国国家研究基金会（NRF）通过教育部资助，资助项目编号为 2019R1A6A1A09031717 和 2021R1A6A3A13046157；部分由韩国政府科学技术信息通信部（MSIT）资助，资助项目编号为 2021R1C1C1006557 和 2021R1I1A3054575；部分由信息通信技术规划与评估研究院（IITP）通过科学技术信息通信部（MSIT）资助，项目包括“基于 5G 新空口的 5G 开放智能定义 RAN（ID-RAN）技术”（资助项目编号：2018-0-01659）和“智能 6G 无线接入系统”（资助项目编号：2021-0-00467）。D. Yu 和 S.-H.Park 任职于全北大学电子工程系及未来半导体融合技术研究中心，韩国全州（电子邮箱：{imcreative93, seokhwan}@jbnu.ac.kr）。H. Lee 任职于釜庆国立大学信息通信工程系，韩国釜山（电子邮箱：hlee@pknu.ac.kr）。S.-E. Hong 任职于电子通信研究院未来移动通信研究部，韩国大田 34129（电子邮箱：iptvguru@etri.re.kr）。版权所有 (c) 2015 IEEE。允许个人使用本资料。但如需将本资料用于任何其他目的，必须向 IEEE 发送申请至 pubs-permissions@ieee.org，以取得许可。

索引词

无蜂窝大规模多输入多输出、深度学习、功率控制。

一、引言

无蜂窝大规模多输入多输出技术凭借增强的覆盖能力，被视为下一代无线网络的一种有前景的解决方案。在无蜂窝大规模多输入多输出系统中，中央处理器（CP）通过前传链路管理分布式接入点（AP）的传输策略。为提升系统性能，经典优化方法依靠中央处理器（CP）处的集中式信号处理，通过收集接入点（AP）测量的信道状态信息（CSI） [1]。然而，这会带来难以承受的前传信令开销，因为短期信道状态信息需要频繁更新。为解决这一问题， [2], [3] 提出了基于长期信道状态信息的功率控制方案，其中所有接入点（AP）的发射功率等级仅利用大尺度衰落状态，在中央处理器（CP）处进行集中优化。由于缺少短期信道状态信息，因此采用性能指标（例如可达数据速率）的启发式近似作为目标函数。这导致实际系统性能与其近似值之间存在模型失配，从而造成最优性损失。

这个问题可以通过深度学习（DL）技术的数据驱动优化能力来处理。近期已有研究针对无小区大规模多输入多输出系统提出了基于深度学习的解决方案 [4]–[6]。文献 [4] 研究了前传量化与多天线波束成形的联合优化任务。该研究假设一种理想情况，即 CP 完全知晓由接入点获取的信道向量。CP 利用集中式深度神经网络（DNN），基于全局网络信道状态信息生成优化解决方案。由于前传协调中的高信令开销，这种集中式深度学习结构并不实用。通过分布式深度神经网络架构可以解决这一问题 [4]–[6]。文献 [5] 借助部署于各个接入点的多个深度神经网络模块，实现了分布式功率控制方案。为了增强分布式决策过程，协同学习（CL）策略是必不可少的，它能够让接入点共享本地可用的统计量，例如本地信道状态信息。文献 [5], 制定了一种启发式协调策略，由 CP 向接入点广播人工生成的控制消息。然而，这种合成协同学习策略并非深度神经网络模块的最优方案。为了解决这一问题，文献 [6] 在 CP 处额外部署了一个深度神经网络，以自主生成自组织协调消息。CP 深度神经网络与接入点深度神经网络共同进行训练，以最大化

所需的目标函数。因此，分布式解计算以及前传协调规则均得到联合优化，从而提高了系统容量。现有的深度学习方法采用固定的计算结构，专用于特定的无小区大规模多输入多输出网络。这削弱了网络规模的可扩展性。例如， [4]–[6] 中提出的深度神经网络无法直接应用于具有不同接入点数量的其他配置。因此，我们需要准备多个针对所有可能网络配置进行训练的深度神经网络。

本文提出了一种通用的协作学习框架，可适应任意给定的无小区大规模 MIMO 配置，尤其是接入点数量随机变化的情况。我们通过以去中心化方式优化各接入点的发射功率分配变量，最大化遍历和速率性能。我们首先提出一种解析优化流程，称为协作随机梯度下降（CSGD）算法。基于样本均值近似（SAA）方法 [7], ，该方法使每个接入点都能通过随机梯度下降方法单独学习有效的解决方案，但需要进行迭代式前传协作。因此，尽管 CSGD 有效，但在前传资源有限时并不实用。

为此，我们提出一种新颖的协作学习框架，通过单次前传协调确定高效的功率控制方案。协作处理器和接入点的去中心化计算由各自的深度神经网络单元完成。这些组件深度神经网络负责生成协作处理器与接入点之间共享的通信消息，并确定各接入点的发射功率等级。为建立可随接入点数量扩展的通用协作学习结构，我们采用参数复用技术，在所有接入点间使用相同的深度神经网络模块。此外，我们精心设计了组件深度神经网络的架构，使其能够处理任意输入/输出维度，不受接入点数量的影响。因此，所提出的协作学习框架可普遍应用于任意给定的无蜂窝大规模 MIMO 网络。数值结果证实，所提出的协作学习框架优于现有深度学习方法。

II. 系统模型与场景

考虑一个无蜂窝大规模多输入多输出系统，其中协作处理器 CP 管理 M 单天线接入点[1] ，与 K 单天线用户设备（UE）进行通信。我们定义索引集

[1]尽管本文在简化分析中假设采用单天线接入点场景，但将其扩展到多天线接入点场景十分直接。多天线接入点的影响也在第五节中进行了讨论。

$\mathcal{K}$, $\{1,2,\ldots,K\}$和 $\mathcal{M}$, $\{1,2,\ldots,M\}$分别表示用户设备和接入点的索引集。令$h_{k,i} \sim \mathcal{CN}(0,\rho_{k,i})$表示接入点 i 与用户设备 k 之间的信道系数，其中$\rho_{k,i}$, $\mathbb{E}[|h_{k,i}|^2]$是对应链路的长期路径损耗。通过采用标准信道获取过程，每个接入点 i都会获得实际信道系数$\mathbf{h}_i$, $\{h_{k,i}\}_{k\in\mathcal{K}}$的本地信道状态信息估计值$\hat{\mathbf{h}}_i$, $\{\hat{h}_{k,i}\}_{k\in\mathcal{K}}$。这里，$h_{k,i}$建模为 [1], [8]

$$h_{k,i} = \hat{h}_{k,i} + e_{k,i}, \tag{1}$$

其中，$e_{k,i}$表示估计误差。采用线性最小均方误差（LMMSE）估计器时，$e_{k,i}$与标称信道$h_{k,i}$不相关。由 [9], 已知，$h_{k,i}$和$e_{k,i}$服从复高斯分布

$$\hat{h}_{k,i} \sim \mathcal{CN}(0,(1-\phi)\rho_{k,i}) \text{ and } e_{k,i} \sim \mathcal{CN}(0,\phi\rho_{k,i}) \tag{2}$$

其中，$\phi \in [0,1]$表示误差比。误差比取决于导频符号 [9]的信噪比。因此，可以将其视为一个根据传播环境动态变化的随机数。

针对无小区大规模多输入多输出系统，已有多种强大的传输策略，例如迫零（ZF）和正则化迫零（RZF）波束成形方法[11]。为实现这些集中式干扰管理方案，接入点应通过前传协调与协作处理器共享其本地信道状态信息估计值$\{\mathbf{h}_i\}_{i\in\mathcal{M}}$。然而，频繁更新短期信道状态信息会产生高昂的前传开销。一种实用的解决方案是让每个接入点 i 将其本地长期信道状态信息 $\boldsymbol{\rho}_i = \{\rho_{k,i}\}_{k\in\mathcal{K}}$ 转发给协作处理器 [2], [3]。这样，协作处理器仍可利用大尺度衰落来抑制多用户干扰，同时降低前传协调中的信令开销 [10]。

因此，本文考虑在各个接入点采用分散式波束成形策略，仅利用本地短期信道状态信息来确定其波束权重。具体而言，采用可通过去中心化方式计算的共轭波束成形（CB）方案，从而实现经济高效的前传协调 [3], [12]。接入点 i 的发射信号 x_i 为

$$x_i = \sum\nolimits_{k\in\mathcal{K}} \sqrt{p_{k,i}} \frac{\hat{h}_{k,i}^*}{|\hat{h}_{k,i}^*|} s_k, \ i \in \mathcal{M}, \tag{3}$$

其中，s_k表示发送给用户设备 k 的数据信号，$p_{k,i}$表示接入点 i 分配给用户设备 k的发射功率。接入点 i的发射功率定义为$\mathbf{p}_i$, $\{p_{k,i}\}_{k\in\mathcal{K}}$，受以下条件约束:

接入点功率预算 P 为 $\sum_{k\in\mathcal{K}} p_{k,\ i} \leq P\ i \in \mathcal{M}$。用户设备 k的遍历速率 R_k 是对 (h, e $\boldsymbol{\rho}$) 的联合分布取平均得到的，可写为

$$R_k = \mathbb{E}_{\hat{\mathbf{h}},\mathbf{e},\boldsymbol{\rho}}\left[\log_2\left(1+\gamma_k(\hat{\mathbf{h}},\mathbf{e},\mathbf{p})\right)\right], \tag{4}$$

其中，p $\{\mathbf{p}_i\}_{i\in\mathcal{M}}$、e $\{e_{k,i}\}_{k\in\mathcal{K},i\in\mathcal{M}}$，而 γ_k(ˆh,e,p) 表示定义如下的信干噪比（SINR）：

$$\gamma_k(\hat{\mathbf{h}},\mathbf{e},\mathbf{p}) = \frac{\left|\sum_{i\in\mathcal{M}} h_{k,i}\hat{h}^*_{k,i}\sqrt{p_{k,i}}/|\hat{h}^*_{k,i}|\right|^2}{1+\sum_{l\in\mathcal{K}\setminus\{k\}}\left|\sum_{i\in\mathcal{M}} h_{k,i}\hat{h}^*_{l,i}\sqrt{p_{l,i}}/|\hat{h}^*_{l,i}|\right|^2}. \tag{5}$$

目标是针对每个给定的实现 (h,e $\boldsymbol{\rho}$)，通过优化发射功率 ˆ p 来最大化平均和速率性能。相应的问题表示为:

$$\max_{\mathbf{p}}.\ \mathbb{E}_{\hat{\mathbf{h}},\mathbf{e},\boldsymbol{\rho}}\left[\sum\nolimits_{k\in\mathcal{K}}\log_2\left(1+\gamma_k(\hat{\mathbf{h}},\mathbf{e},\mathbf{p})\right)\right] \tag{6a}$$

$$\text{s.t.}\ \ \sum\nolimits_{k\in\mathcal{K}} p_{k,i} \leq P,\ \forall i\in\mathcal{M}. \tag{6b}$$

问题 (6) 通常是非凸的，因此难以获得全局最优的 ˆ 解。对于任意分布的 CSI (h,e $\boldsymbol{\rho}$)，其期望没有解析表达式，这使得应用传统非凸优化技术颇为困难。传统方法[2], [3] 提出了便于处理的平均速率目标闭式近似。为此，利用詹森不等式直接去除所有小尺度衰落系数，但这会导致遍历速率与其近似值之间出现模型失配。此外，由于近似速率表达式仅取决于长期信道统计信息，因此在优化功率控制变量时无法利用短期 CSI。此外，分散部署的接入点需要一种新颖的去中心化计算架构。每个接入点 i 都应仅根据局部网络知识（即本地信道状态信息向量 $\mathbf{h}_i$ 和 $\boldsymbol{\rho}_i$）推断其本地功率分配解 $\mathbf{p}_i$ 。这种局部观测不足以单独恢复问题 (6) 的最优解。因此，接入点之间的交互对于构建高效的功率控制方案至关重要。

III. 协作随机梯度下降

本节提出一种 CSGD 算法，采用 SAA 方法处理式（6a）中的期望项 [7]。CSGD 无需对和速率目标函数进行闭式近似。因此，CSGD 算法为

现有功率控制方案 [2], [3] 提供了性能上界，这些方案会引入模型失配误差。其核心思想是让每个接入点基于标准小批量随机梯度下降算法，分别更新其本地解 $\mathbf{p}_i$ 。接入点首先通过前传链路共享其本地信道状态信息，尤其是长期信道状态信息 $\boldsymbol{\rho}_i$。根据式（2）中的已知分布，每个接入点 i 都可以生成一个小批量集合 $\mathcal{B}_i$ ，其中包含其本地信道状态信息误差 $\mathbf{e}_i$ ，$\{e_{k,i}\}_{k\in\mathcal{K}}$ ，以及其他接入点统计信息 $\hat{\mathbf{h}}_{-i}$ ，$\{\hat{h}_{k,\ j}\}_{j\in\mathcal{M}\setminus\{i\},\ k\in\mathcal{K}}$ 和 $\mathbf{e}_{-i}$ ，$\{e_{k,j}\}_{j\in\mathcal{M}\setminus\{i\},\ k\in\mathcal{K}}$。设 $\mathbf{b}_i^{(n)}$, $(\mathbf{e}_i^{(n)}, \hat{\mathbf{h}}_{-i}^{(n)}, \mathbf{e}_{-i}^{(n)}) \in \mathcal{B}_i$ 为接入点 i的第 n 个批次样本。此外，我们将其他接入点的发射功率记为 $\mathbf{p}_{-i}$,$\{p_{k,j}\}_{j\in\mathcal{M}\setminus\{i\},\ k\in\mathcal{K}}$ 。随后，每个接入点 i都可以估计遍历和速率（6a）的 SAA，记为 $R_i(\mathbf{p}_i, \mathbf{p}_{-i})$，其表达式为

$$\bar{R}_i(\mathbf{p}_i, \mathbf{p}_{-i}) = \frac{1}{|\mathcal{B}_i|}\sum\nolimits_{k\in\mathcal{K}}\sum\nolimits_{\mathbf{b}_i^{(n)}\in\mathcal{B}_i}\log_2(1+\bar{\gamma}_{k,i}(\hat{\mathbf{h}}_i, \mathbf{b}_i^{(n)}, \mathbf{p}_i, \mathbf{p}_{-i})), \tag{7}$$

其中，$\bar{\gamma}_{k,\ i}(\hat{\mathbf{h}}_i,\mathbf{b}_i^{(n)},\mathbf{p}_i,\mathbf{p}_{-i})$ 表示在接入点 i 处针对其批次样本 $\mathbf{b}_i^{(n)}$ 计算的信干噪比，定义见下一页顶部的式 (9)。在式 (7) 中，未知统计量（如信道状态信息误差 $\mathbf{e}_i$ 以及其他接入点信息 $\mathbf{h}_{-i}$ 和 $\mathbf{e}_{-i}$ ）均在小批次样本上取平均。这样，我们便可通过基于小批次的样本平均近似来估计遍历和速率。在计算出 $R_i(\mathbf{p}_i,\mathbf{p}_{-i})$后，每个接入点 i 分别执行投影随机梯度下降更新：

$$\mathbf{p}_i \leftarrow \mathcal{P}\left[\mathbf{p}_i + \alpha\nabla_{\mathbf{p}_i}\bar{R}_i(\mathbf{p}_i, \mathbf{p}_{-i})\right], \tag{8}$$

其中，∇_X 表示关于变量 X的梯度算子，$\alpha > 0$ 表示每次更新的步长，$\mathcal{P}[\cdot]$ 表示投影到可行集 (6b) 上的投影算子。在式 (8) 中，来自其他接入点的解 $\mathbf{p}_{-i}$（可由其他接入点提供）被简单地视为常量。

我们在算法1中总结了CSGD方法。在每次迭代中，接入点首先在协作处理器的帮助下，通过前传信道交换各自的本地统计量，即本地长期信道状态信息 $\boldsymbol{\rho}_i$ 和本地解 $\mathbf{p}_i$。然后，每个接入点根据式（8）更新其功率控制向量 $\mathbf{p}_i$ 。重复此过程直至收敛。由于更新规则（8）仅利用本地信息即可执行，因此CSGD能够以去中心化方式实现。CSGD方法的计算复杂度主要由梯度计算决定。为计算 $\nabla_{\mathbf{p}_i}\bar{R}_i(\mathbf{p}_i, \mathbf{p}_{-i})$，每个接入点 i需要进行 $\mathcal{O}(|\mathcal{B}_i|MK^2)$ 次计算，因为在式（7）的样本平均近似中使用了 $|\mathcal{B}_i|$ 个批次样本。令 L 表示CSGD的迭代次数，则总复杂度为 $\mathcal{O}(L|\mathcal{B}_i|MK^2)$。

CSGD算法的迭代特性不仅需要反复共享 $\mathbf{p}_i$ ，导致前传开销过高，还会增加计算复杂度。为应对这一挑战，我们提出一种基于深度学习的方法，该方法只需一次性协调。

$$\bar{\gamma}_{k,i}(\hat{\mathbf{h}}_i, \mathbf{b}_i^{(n)}, \mathbf{p}_i, \mathbf{p}_{-i}) = \frac{\left|(\hat{h}_{k,i}+e_{k,i}^{(n)})\sqrt{p_{k,i}}\hat{h}_{k,i}^{*}/|\hat{h}_{k,i}^{*}|+\sum_{j\in\mathcal{M}\setminus\{i\}}(\hat{h}_{k,j}^{(n)}+e_{k,j}^{(n)})\sqrt{p_{k,j}}\hat{h}_{k,j}^{*(n)}/|\hat{h}_{k,j}^{*(n)}|\right|^2}{1+\sum_{l\in\mathcal{K}\setminus\{k\}}\left|(\hat{h}_{k,i}+e_{k,i}^{(n)})\sqrt{p_{l,i}}\hat{h}_{l,i}^{*(n)}/|\hat{h}_{l,i}^{*(n)}|+\sum_{j\in\mathcal{M}\setminus\{i\}}(\hat{h}_{k,j}^{(n)}+e_{k,j}^{(n)})\sqrt{p_{l,j}}\hat{h}_{l,j}^{*(n)}/|\hat{h}_{l,j}^{*(n)}|\right|^2} \tag{9}$$

算法1 CSGD算法

1: 重复

2: 每个接入点 i 通过前传链路将其本地解 $\mathbf{p}_i$ 共享给其他接入点 $\mathcal{M}\setminus\{i\}$。

3: 每个接入点 i 从 (2) 中采样一个小批量集合，并计算 $\bar{R}_i(\mathbf{p}_i, \mathbf{p}_{-i})$ 在 (7) 中。

4: 每个接入点 i 根据 (8) 更新其本地解 $\mathbf{p}_i$。

5: 直至收敛

IV. 提出的协作学习方法

本节提出了一种协作学习框架，通过分布式CP-AP协作优化发射功率控制变量。CP和接入点均配备了各自的深度神经网络模块，用于执行相应实体的协调与计算。我们首先设计协作式深度神经网络推理，随后采用联合训练策略。

A. 协作学习机制

如图1所示，所提出的协作学习框架由三个依次进行的步骤组成：上行前传协作、下行前传协作和分散式决策。各步骤的详情

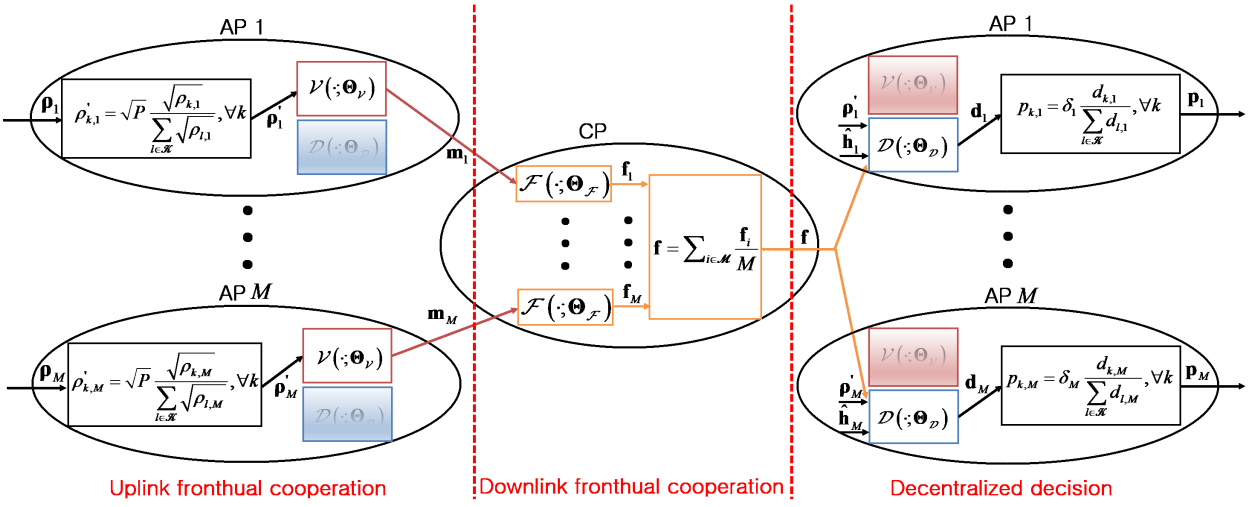


图1：所提出的协作学习结构

如下所述。

1) 上行前传协作：对于协同优化，协作处理器需要收集来自各接入点的本地信息。为此，每个接入点 i 首先通过前传链路向协作处理器发送一条长度为 d_U 的上行链路消息 $\mathbf{m}_i \in \mathbb{R}^{d_U}$。该消息基于本地长期信道状态信息$\boldsymbol{\rho}_i$ 生成，具体如下：

$$\mathbf{m}_i = \mathcal{V}_i(\boldsymbol{\rho}_i'; \boldsymbol{\Theta}_{\mathcal{V}_i}), \tag{10}$$

其中， $\mathcal{V}_i(\cdot; \boldsymbol{\Theta}_{\mathcal{V}_i})$ 表示接入点 i 处具有可训练参数 $\boldsymbol{\Theta}_{\mathcal{V}_i}$ 的消息生成深度神经网络。深度神经网络 $\mathcal{V}_i(\cdot; \boldsymbol{\Theta}_{\mathcal{V}_i})$ 的输入特征 $\boldsymbol{\rho}_i'$, $\{\rho_{k,i}'\}_{k\in\mathcal{K}}$ 按如下方式获取：

$$\rho_{k,i}' = \sqrt{P}\frac{\sqrt{\rho_{k,i}}}{\sum_{l\in\mathcal{K}}\sqrt{\rho_{l,i}}}. \tag{11}$$

(11) 中的数据预处理对长期本地信道状态信息 $\boldsymbol{\rho}_i$ 进行归一化，使得到的输入特征 $\boldsymbol{\rho}_i'$ 位于有界区域 $\rho_{k,i}' \in [0, \sqrt{P}]$ 内。这将深度神经网络的输入限制在一个紧凑集合中，从而满足通用逼近定理 [13] 的必要条件。因此，通过使深度神经网络关注各个信道链路 $\rho_{k,i}$ 的重要性，而非其绝对值，可以加快训练过程。归一化后的长期信道状态信息有助于无小区大规模多输入多输出系统 [5], [14] 的优化。

如式 (10) 所示，我们可以采用专用于每个接入点 $i \in \mathcal{M}$ 的独立深度神经网络 $\mathcal{V}_i(\cdot; \boldsymbol{\Theta}_{\mathcal{V}_i})$。然而，这种方法无法灵活适应接入点数量 M 的变化。在某个 M 经过训练的深度神经网络组 $\{\mathcal{V}_i(\cdot; \boldsymbol{\Theta}_{\mathcal{V}_i})\}_{i\in\mathcal{M}}$ ，无法直接应用于具有不同 M 的其他网络配置。这就需要额外的训练步骤，为所有可能的无小区大规模多输入多输出配置准备多个深度神经网络实例。这种情况常见于动态无小区大规模多输入多输出系统，其中只有部分接入点处于激活状态，以实现能效通信 [15]。为解决这一问题，我们采用一种可扩展学习架构，使深度神经网络的计算不受接入点数量 M 的影响。对于所有接入点，我们复用相同的深度神经网络 $\mathcal{V}(\cdot; \boldsymbol{\Theta}_{\mathcal{V}})$，以实现式 (10) 中用于生成上行链路消息的推理，具体如下

$$\mathbf{m}_i = \mathcal{V}(\boldsymbol{\rho}_i'; \boldsymbol{\Theta}_{\mathcal{V}}). \tag{12}$$

这种参数共享方法已广泛应用于现有的深度神经网络架构中，例如卷积神经网络（CNN）和图神经网络。这对于提升深度神经网络的泛化能力尤为有益，使其能够适应未见过的输入分布，例如在训练期间不可用的新系统规模 M 。

2）下行前传协作：协作处理器（CP）使用其深度神经网络 $\mathcal{F}(\cdot;\ \boldsymbol{\Theta}_{\mathcal{F}})$ 和参数集 $\boldsymbol{\Theta}_{\mathcal{F}}$，对分区的信息向量 $\{\mathbf{m}_i\}_{i\in\mathcal{M}} \in \mathbb{R}^{Md_U}$进行聚合。相应的输出记为 $\mathbf{f} \in \mathbb{R}^{d_D}$，长度为 d_D，它作为下行通信消息广播给所有接入点。一种简单的方法是将所有上行消息拼接起来作为输入特征，即 $\mathbf{f} = \mathcal{F}(\{\mathbf{m}_i\}_{i\in\mathcal{M}};\ \boldsymbol{\Theta}_{\mathcal{F}})$ [6]。然而，由于输入维度会随着 M 增大，这种方法无法适应接入点数量 M 的扩展。

为了解决这一问题，我们在 CP 处构建了维度不变推理，使下行链路消息 $\mathbf{f}$ 的计算过程与 M 无关。首先，我们从上行链路消息 $\mathbf{m}_i$ 中提取潜在特征 $\mathbf{f}_i \in \mathbb{R}^{d_D}$，如下所示：

$$\mathbf{f}_i = \mathcal{F}\left(\mathbf{m}_i; \boldsymbol{\Theta}_{\mathcal{F}}\right). \tag{13}$$

与式（10）类似，式（13）中也采用了参数共享的概念，使得单个 DNN $\mathcal{F}(\cdot;\ \boldsymbol{\Theta}_{\mathcal{F}})$ 能够并行生成一组信息向量 $\{\mathbf{f}_i\}_{i\in\mathcal{M}}$。利用非正交多址系统中的叠加编码概念，将下行链路消息向量 $\mathbf{f}$ 设计为 $\mathbf{f}_i$、$\forall i \in \mathcal{M}$ 的平均值，即：

$$\mathbf{f} = \frac{1}{M}\sum\nolimits_{i\in\mathcal{M}} \mathbf{f}_i = \frac{1}{M}\sum\nolimits_{i\in\mathcal{M}} \mathcal{F}\left(\mathbf{m}_i; \boldsymbol{\Theta}_{\mathcal{F}}\right). \tag{14}$$

请注意，式（14）可视为平均池化操作，该操作已广泛应用于卷积神经网络中。通过从本地接入点消息向量 $\{\mathbf{f}_i\}_{i\in\mathcal{M}}$ 中剔除不必要的统计量，该操作有助于提取重要的全局特征 $\mathbf{f}$。因此，我们能够高效地实现维度不变的前传协作。

3) 分散式决策：为确定本地功率分配解 $\mathbf{p}_i$，每个接入点 i^ 利用从协作处理器接收的下行消息 $\mathbf{f}$，以及其本地信道状态信息 $\boldsymbol{\rho}_i'$ 和 $\mathbf{h}_i$。所有接入点的这一分布式决策过程由深度神经网络 $\mathcal{D}(\cdot;\boldsymbol{\Theta}_{\mathcal{D}})$ 建模，其可训练参数为 $\boldsymbol{\Theta}_{\mathcal{D}}$。因此，每个接入点 i 恢复其解 $\mathbf{p}_i$，如下所示

$$\mathbf{p}_i = \mathcal{D}(\mathbf{f}, \boldsymbol{\rho}_i', \hat{\mathbf{h}}_i; \boldsymbol{\Theta}_{\mathcal{D}}). \tag{15}$$

由于 $\mathcal{D}(\cdot;\boldsymbol{\Theta}_{\mathcal{D}})$ 的输出会直接用作发射功率变量，因此需要合理设计输出层，以保证满足功率约束 (6b)。为此，我们在输出层设计了一种新型激活函数。设 $\mathbf{d}_i = [d_{1,i} \cdots d_{K,i}, \delta_i] \in \mathbb{R}^{K+1}$ 为应用激活函数前 $\mathcal{D}_i(\cdot, \boldsymbol{\Theta}_{\mathcal{D}})$ 的输出向量。前 K 个元素 $d_{k,i} \geq 0$, $k \in \mathcal{K}$用于控制发射功率变量 $p_{k,i}$, $k \in \mathcal{K}$之间的比例。相反，最后一个元素

δ_i 决定接入点 i所消耗的总发射功率。为了将 δ_i 限制在可行范围 $[0, P]$内，我们采用修正线性单元6（ReLU6）函数，定义如下：

$$\delta_i \leftarrow P \min\left(\max\left(\delta_i, 0\right), 6\right)/6. \tag{16}$$

功率控制变量 $p_{k,\ i}$ 随后可求得

$$p_{k,i} = \frac{\delta_i d_{k,i}}{\sum_{l\in\mathcal{K}} d_{l,i}}, \tag{17}$$

这总是会得到可行解 $\sum_{k\in\mathcal{K}} p_{k,\ i} = \delta_i \leq P$。深度神经网络 $\mathcal{D}(\cdot; \boldsymbol{\Theta}_{\mathcal{D}})$ 的输出激活可以通过式（16）和（17）中的运算来指定。

最后，式（12）、（14）和（15）中的一组深度神经网络为所提出的协作学习提供了端到端的前向传播映射$\mathcal{G}(\cdot;\ \boldsymbol{\Theta})$，即 $\mathbf{p} = \mathcal{G}(\boldsymbol{\rho}, \hat{\mathbf{h}};\ \boldsymbol{\Theta})$，其中 $\boldsymbol{\Theta}$，$\{\boldsymbol{\Theta}_{\mathcal{V}}, \boldsymbol{\Theta}_{\mathcal{F}}, \boldsymbol{\Theta}_{\mathcal{D}}\}$表示所有可训练参数的集合。

B. 训练与实现

我们讨论所提出的协作学习架构的联合训练过程。将 $\mathbf{p} = \mathcal{G}(\boldsymbol{\rho}, \mathbf{h};\ \boldsymbol{\Theta})$ 代入（6），可得训练问题如下：

$$\max_{\boldsymbol{\Theta}}.\ \mathbb{E}_{\hat{\mathbf{h}},\mathbf{e},\boldsymbol{\rho}}\left[\sum_{k\in\mathcal{K}} \log_2\left(1+\gamma_k\left(\hat{\mathbf{h}}, \mathbf{e}, \mathcal{G}(\boldsymbol{\rho}, \hat{\mathbf{h}}; \boldsymbol{\Theta})\right)\right)\right], \tag{18}$$

其中，功率约束 (6b) 被移除，因为根据 (16) 和 (17)，该约束始终满足。(18) 中的训练问题可以通过现有的小批量随机梯度下降算法轻松解决，例如 Adam优化器。训练数据集包含大量长期信道状态信息 $\boldsymbol{\rho}$ 的实现。在每个训练轮次中，我们随机抽取一个由长期信道状态信息$\boldsymbol{\rho}$组成的小批量集合。这些信息可以预先通过实验收集，也可以根据已知的接入点-用户设备部署场景生成。随后，我们利用已知分布 (2) 生成信道状态信息估计值 $\mathbf{h}$ 和误差向量 $\mathbf{e}$。由于误差比 ϕ 在实际中会随机变化，因此有必要构建能够适应任意变化的 ϕ 的通用深度神经网络。为此，我们在训练步骤中随机生成误差比因子，例如从均匀分布 $\phi \sim \mathcal{U}(0, 1)$ 中生成。因此，所提出的协作学习可以普遍应用于任意信道状态信息误差统计 ϕ。我们利用这些信息计算在小批量集合上取平均的训练目标 (18) 的梯度。因此，所提出的协作学习通过观测大量人工生成的信道状态信息误差样本进行训练。通过这种方式，深度神经网络可以根据实际信道状态信息的估计值学习其未知分布，从而生成稳健的功率控制机制。

所提出的联合训练过程以离线方式实现，方法是将所有组件深度神经网络集中到协作处理器。训练后的深度神经网络模块被部署到所需的网络实体，以进行去中心化功率优化。在这一实现阶段，我们不再需要信道状态信息误差 e，因为所提出的协作学习 $\mathcal{G}(\boldsymbol{\rho}, \mathrm{h}; \Theta)$ 仅接收长期信道状态信息 $\boldsymbol{\rho}$ 和短期信道状态信息估计值 h。

接入点的数量 M 可视为所提出协作学习策略的一个超参数。令 M_{train} 表示训练步骤中采用的接入点数量。为进一步提高可扩展性，应谨慎选择M_{train}，使得在特定 M_{train} 下训练的所提出协作学习能够在较宽范围的测试接入点数量 M_{test}上普遍良好地工作。该问题将在第五节中阐明。

V. 数值结果

本节验证所提出的协作学习方法。接入点和用户设备均匀分布在半径为300 m的圆形区域内。长期信道状态信息建模为$\rho_{k,\,i} = P_0(q_{k,\,i}/q_0)^{-\eta}$，其中$q_{k,\,i}$ 是接入点 i 与用户设备 k之间的距离，$P_0 = 10$ 是在参考距离 $q_0 = 30$ m处的路径损耗，$\eta = 3$ 表示路径损耗指数。所有组件深度神经网络均采用16个全连接隐藏层实现。所有隐藏层的输出维度均设置为$160K$。所有隐藏层均采用ReLU激活函数，后接批归一化层。消息维度固定为 $d_U = d_D = K$。在 2×10^5 个测试样本上评估训练后的深度神经网络性能。

我们考虑以下基准方案。

- CSGD算法：算法1中的CSGD方法为所提出的协作学习框架生成一个无法实现的性能上界，因为该方法允许接入点迭代地共享功率控制解 $\mathrm{p}_i, \forall i \in \mathcal{M}$。

- 非合作学习（NCL）：该方案不允许接入点之间进行任何消息交换，即 $d_U = d_D = 0$, 因此给出了所提出的协调机制的性能下界。

- 合成协作学习（SCL） *[5]*：该方法依赖于一种人工协作策略。上行链路消息 m_i 被简单地固定为本地长期信道状态信息 $\mathrm{m}_i = \boldsymbol{\rho}_i$。此外，下行链路消息向量 f_i 的第 k 个元素 $f_{k,\,i}$ 由归一化的本地长期信道状态信息给出，即 $f_{k,\,i} = \sqrt{P}\rho_{k,\,i}/\Sigma_{j\in\mathcal{M}}\rho_{k,\,j}$。

- 等功率分配：每个接入点 i 对所有用户设备采用等功率分配 $p_{k,\ i} = P/K,\ \ \forall k \in \mathcal{K}$。

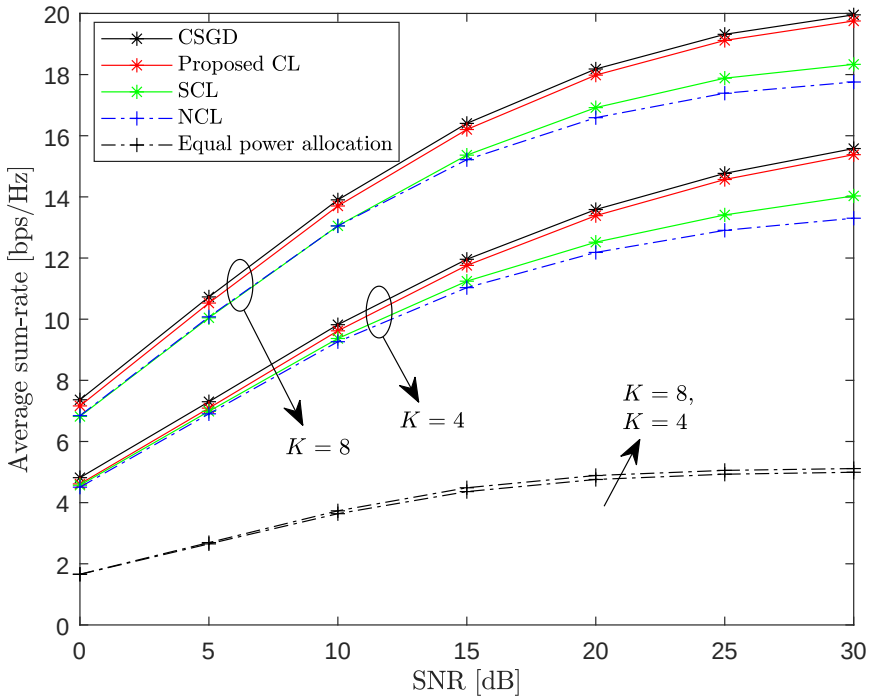
请注意， [5] 中的SCL基线不采用参数共享策略，即每个接入点都有专属的深度神经网络模块。因此，这些基线需要针对每种接入点数量 M，分别准备多个训练好的深度神经网络实例。相反，所提出的可扩展方法在所有可能的 M中均使用同一组深度神经网络。

图2展示了随信噪比变化的平均和速率性能，其中信噪比定义为P。图2(a)表明，与CSGD算法相比，所提出的协作学习方法造成的性能损失可以忽略不计。这验证了所提方法能够达到性能上界的有效性。在所有仿真场景中，所提出的协作学习方法均优于深度学习基线。在高信噪比区域，性能增益尤为显著，因为此时用户间干扰主导了整个系统容量。这说明，所提出的前传协调策略对于协作处理器缓解干扰至关重要。

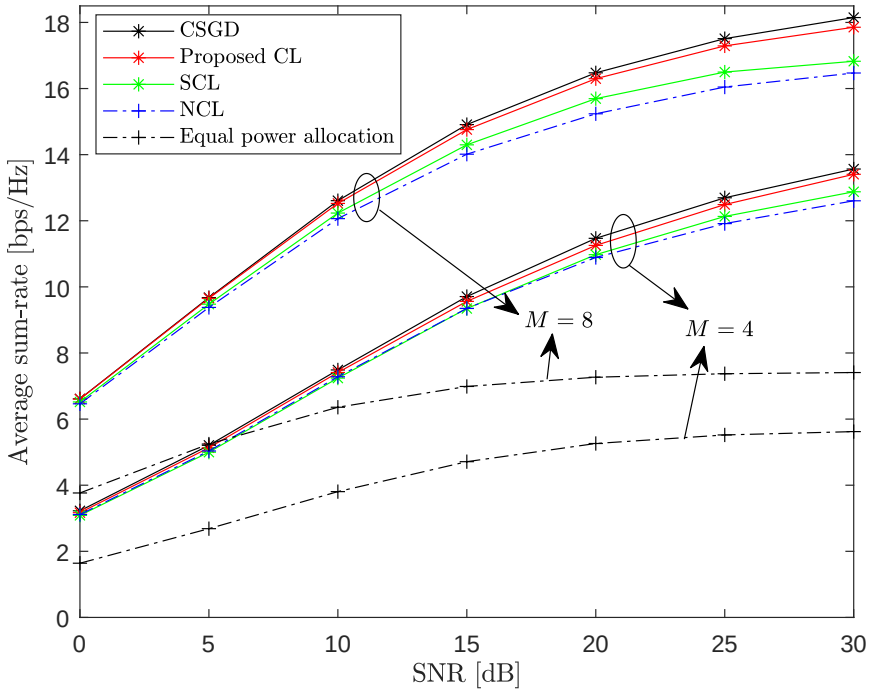
图2(b)绘制了多天线接入点场景下的平均和速率性能，其中每个接入点有两根天线。只需对式(3)中的发射信号x_i 稍作修改，即可将所提出的CSGD和协作学习方法直接应用于这一多天线接入点场景。这里，向量值信号$\mathbf{x}_i \in \mathbb{C}^N$ 具有 N 根接入点天线，其中$\mathbf{x}_i = \sum_{k\in\mathcal{K}} \sqrt{p_{k,\ i}}\hat{\mathbf{h}}_{k,\ i}s_k/||\hat{\mathbf{h}}_{k,\ i}||$，其中$\hat{\mathbf{h}}_{k,\ i} \in \mathbb{C}^N$表示从接入点 i 到用户设备 k的估计信道状态信息向量。从图中可以看出，所提出的协作学习方法在多天线接入点场景下仍表现良好，性能几乎与CSGD性能上界相同。因此，我们可以得出结论：所提出的协作学习方法可应用于任意数量的接入点天线。

图3绘制了 ϕ 下， $M = 8$、 $K = 4$ 和信噪比为 = 20 dB时的平均和速率与误差比的关系。为验证所提方法的鲁棒性，我们还在图3中评估了一种非鲁棒方法，其中深度神经网络仅使用实际信道进行训练，即信道状态信息估计误差 e 未纳入训练过程。与其他方案相比，所提方案性能更优，同时与 CSGD 算法相比损失可忽略不计。随着误差比增大，所提协作学习方法带来的增益愈发显著。所提协作学习方法将前传协调纳入优化域。因此，经过优化的前传消息可作为接入点处分布式决策深度神经网络的充分统计量。然而，当信道状态信息误差较大时 $\phi = 0.5$，合成协作学习的表现优于非鲁棒协作学习，这表明在信道状态信息误差较大的情况下，鲁棒训练策略至关重要。

图4展示了　　所提出的协作学习根据采样方式得到的平均和速率性能



(a) 单天线接入点， $M = 8$



(b) 多天线接入点， $N = 2$, $K = 4$

图2： $M \in \{4, 8\}$, $K \in \{4, 8\}$ 和 $\phi = 0.1$对应的平均和速率随信噪比的变化。

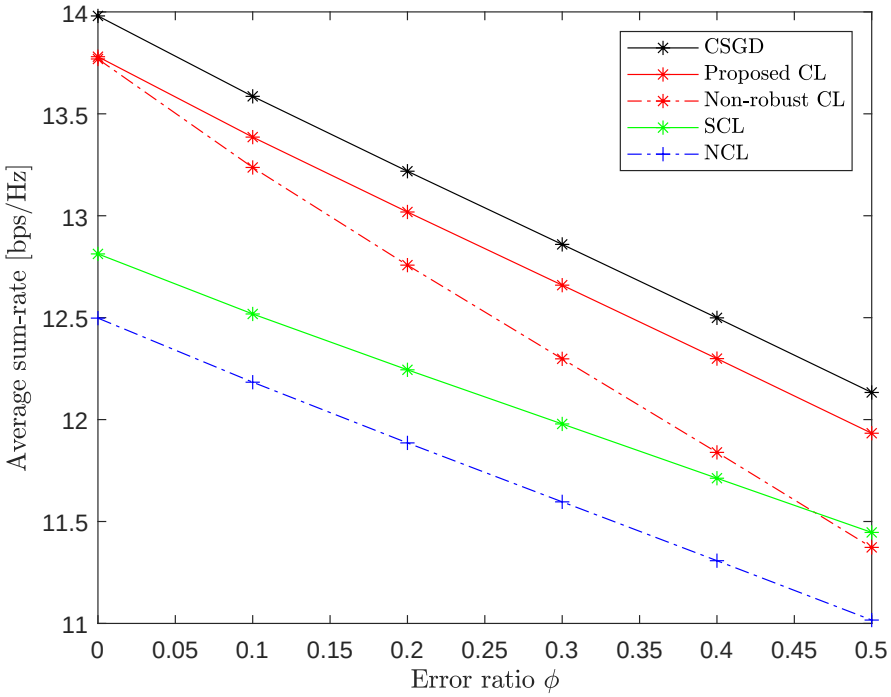


图 3: 对于 $M = 8$、 $K = 4$ 和信噪比 = 20 d, 平均和速率随误差比 ϕ 的变化B

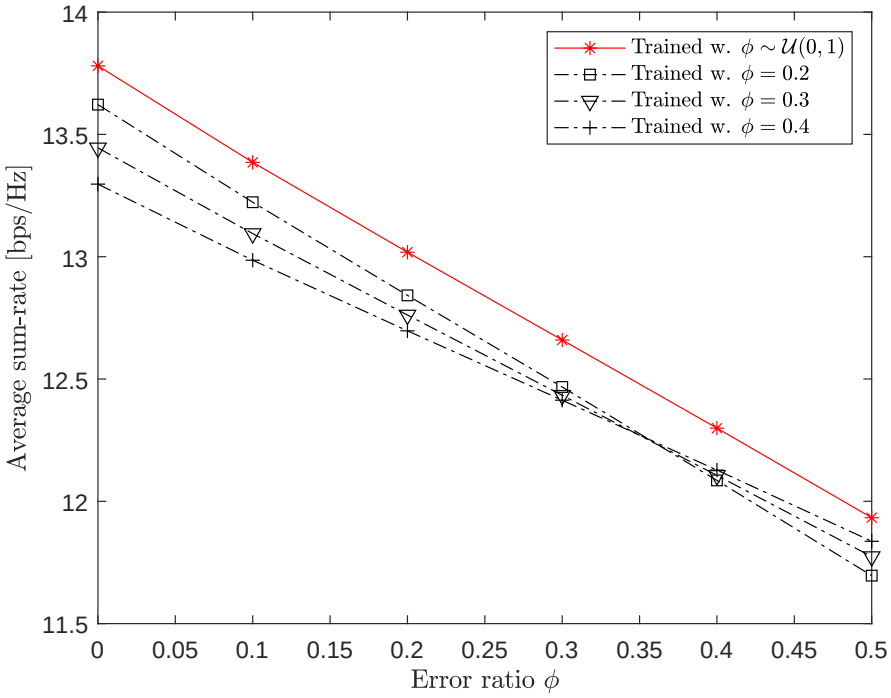


图 4: 对于 $M = 8$、 $K = 4$ 和信噪比 = 20 dB, 平均和速率随误差比 ϕ 的变化

表 I： $K = 16$、 $\phi = 0.1$ 和信噪比为 $= 20$ dB 时的相对和速率

M_{train} \ M_{test}	16	20	24	28	32
8	0.93	0.93	0.92	0.94	0.92
16	0.94	0.95	0.94	0.96	0.96
24	0.93	0.96	0.99	0.97	0.97
32	0.93	0.97	0.97	0.98	0.99

误差比为 ϕ 时， $M = 8$、 $K = 4$ 和信噪比为 = 20 dB。由图可见，当训练 ϕ 与测试 ϕ 不同时，和速率性能会下降。根据结果可知，使用准确的 ϕ 进行训练对于所提出协作学习的鲁棒优化至关重要。此外，即使在误差比完全已知的区域，所提出的协作学习使用从 $\phi \sim \mathcal{U}(0, 1)$ 中随机采样的 ϕ 进行训练，其性能也优于使用固定 ϕ 进行训练的协作学习。这一结果表明，我们的鲁棒训练策略通过随机采样 ϕ，不仅可以避免在线阶段缺乏准确长期统计信息所导致的性能下降，还可以通过获得诸如噪声注入[16]之类的正则化效应来提升性能。

接下来，我们通过在训练和测试步骤中采用不同数量的接入点 M ，来考察所提出的协作学习方法的灵活性和可扩展性。具体而言，在 M_{train} 个接入点下训练的协作学习方法，可直接应用于包含 M_{test} 个接入点的无蜂窝大规模多输入多输出系统。我们评估相对和速率性能，该指标定义为所提方案达到的和速率与 CSGD 方法达到的和速率之比。表 I 列出了所提出的协作学习方法在 $M_{\text{train}} \in \{8, 16, 24, 32\}$ 下训练、在 $M_{\text{test}} \in \{16, 20, 24, 28, 32\}$ 下测试时，于 $K = 16$、 $\phi = 0.1$ 和信噪比为 = 20 dB 条件下的相对和速率性能。总体而言，所提出的协作学习方法在所有仿真的 M_{test} 中都表现良好，即使是在接入点数量达到 $M = 32$ 的大规模场景下也是如此，尽管训练步骤中未出现过其他配置 $M_{\text{test}} \neq M_{\text{train}}$ 。因此，我们只需针对优化后的超参数 M_{train} 执行一次训练过程。与现有方法 [5], [6] 相比，这大幅降低了训练复杂度，因为后者必须针对所有可能的 M_{test} 进行训练。

六、结论

本文提出了一种基于深度学习的无小区大规模多输入多输出系统分布式功率控制策略。本文还提出了一种新颖的协作学习结构，通过组件深度神经网络实现前传协作和

去中心化决策过程。因此，协作处理器和接入点可以通过交换面向深度神经网络的通信消息，确定适当的功率控制机制。本文采用参数共享的概念，使所提出的协作学习的前向传播计算不受接入点数量的影响。数值结果证明了所提方法的有效性。将该方法扩展至多天线用户设备场景值得作为未来工作加以研究。

参考文献

[1] J. Choi、N. Lee、S.-N. Hong 和 G. Caire，“不完美发射端信道状态信息下多小区多用户多输入多输出下行系统的联合用户选择、功率分配与预编码设计”，IEEE无线通信汇刊，第19卷，第1期，第162–176页，2020年1月。[2] E. Nayebi、A. Ashikhmin、T. L. Marzetta、H. Yang 和 B. D. Rao，“无蜂窝大规模多输入多输出系统中的预编码与功率优化”，IEEE无线通信汇刊，第16卷，第7期，第4445–4459页，2017年7月。[3] H. Q. Ngo、A. Ashikhmin、H. Yang、E. G. Larsson 和 T. L. Marzetta，“无蜂窝大规模多输入多输出与小小区的比较”，IEEE无线通信汇刊，第16卷，第3期，第1834–1850页，2017年3月。[4] D. Yu、H. Lee、S.-H. Park 和 S.-E. Hong，“云无线接入网络中波束成形与前传量化联合优化的深度学习方法”，IEEE无线通信快报，第10卷，第10期，第2180–2184页，2021年10月。[5] M. Zaher、O. T. Demir、E. Björnson 和 M. Petrova，“无蜂窝大规模多输入多输出系统中基于学习的下行功率分配”，IEEE无线通信汇刊，第22卷，第1期，第174-188页，2023年1月。[6] H. Lee、J. Kim 和 S.-H. Park，“雾无线接入网络中的最优前传学习与去中心化边缘计算”，IEEE无线通信汇刊，第20卷，第9期，第5599–5612页，2021年9月。[7] O. Shamir 和 N. Srebro，“分布式随机优化与学习”，载于《年度Allerton通信、控制与计算会议论文集》（Allerton），2015年2月。[8] H. Yin、D. Gesbert、M. Filippou 和 Y. Liu，“大规模多天线系统中信道估计的协同方法”，IEEE通信领域精选期刊，第31卷，第2期，第264–273页，2013年2月。[9] B. Nosrat-Makouei、J. G. Andrews 和 R.W.Heath，“不完美信道状态信息下相关信道上的多输入多输出干扰对齐”，IEEE信号处理汇刊，第59卷，第6期，第2783–2794页，2011年6月。[10] G. Dartmann、M. Jordan、X. Gong 和 G. Ascheid，“多用户多小区单播场景中采用长期波束成形和低信干噪比反馈速率的小区间干扰抑制”，载于《IEEE车辆技术会议论文集》（VTC），2009年5月。[11] F. Riera-Palou 和 G. Femenias，“采用迫零的无蜂窝大规模多输入多输出中的去中心化问题”，载于《年度Allerton通信、控制与计算会议论文集》（Allerton），2019年9月。[12] H. Q. Ngo、A. Ashikhmin、H. Yang、E. G. Larsson 和 T. L. Marzetta，“无蜂窝大规模多输入多输出：为每个人提供普惠优质服务”，载于《IEEE信号处理进展与无线通信国际研讨会论文集》（SPAWC），第201–205页，2015年6月。[13] K. Hornik、M. Stinchcombe 和 H. White，“多层前馈网络是通用逼近器”，《神经网络》，第2卷，第5期，第359–366页，1989年1月。[14] G. Interdonato、P. Frenger 和 E. G. Larsson，“无蜂窝大规模多输入多输出的可扩展性方面”，载于《IEEE国际通信会议论文集》（ICC），2009年5月。[15] J. Na、J. Koh、S. Park 和 J. Kang，“通过动态远端无线头选择提升云端与边缘处理的能效”，载于《IEEE消费者通信与网络会议论文集》（CCNC），2018年1月。[16] I. Goodfellow、Y. Bengio 和 A. Courville，《深度学习》。美国马萨诸塞州剑桥：MIT 出版社，2016 年。