DeepFP：用于 MIMO 波束成形的深度展开分式规划

Jianhang Zhu，IEEE 研究生会员；Tsung-Hui Chang，IEEE 会士；Liyao Xiang，IEEE 会员；Kaiming Shen，IEEE 高级会员

arXiv:2601.02822v1 [cs.IT] 6 Jan 2026

摘要——本文针对多输入多输出（MIMO）无线网络中的加权和速率波束成形问题，提出一种结合学习与优化的方法。传统方法，即分式规划（FP）方法和加权最小均方误差（WMMSE）算法，计算开销较大，原因有二：（i）它们需要对一系列矩阵求逆，而这些矩阵的大小与天线数量成正比；（ii）它们需要调整一组拉格朗日乘子，以处理功率约束。近期提出的简化WMMSE方法解决了单小区情形下的上述两个问题。相比之下，对于多小区情形，另一种近期提出的FastFP方法通过采用改进的FP技术，避免了大型矩阵求逆和拉格朗日乘子的使用，但FastFP中的更新步长难以确定。因此，我们提出将深度展开网络融入FastFP，以优化步长。数值实验表明，所提方法比基于加权最小均方误差算法的学习方法效率高得多。

索引词——多输入多输出（MIMO）波束成形、加权和速率最大化、深度展开。

I. 引言

多输入多输出（MIMO）系统设计中的一个基本问题，是优化发射波束成形器，以最大化整个蜂窝网络中的加权和速率（WSR），即WSR问题。加权最小均方误差（WMMSE）算法 [2], [3] 和分式规划（FP） [4], [5] 是该领域的两种常用方法。由于这两种方法都采用迭代结构，一个自然的想法是通过深度展开来学习它们的行为，正如 [6]–[13]中广泛开展的研究。然而，在多小区 MIMO场景下会出现两个主要挑战：（i）迭代算法（例如WMMSE或FP）需要进行大矩阵求逆，但与矩阵加法相比，矩阵求逆更难学习，且

论文于2026年1月3日被IEEE通信汇刊接收。Jianhang Zhu和Kaiming Shen的工作部分获国家自然科学基金委员会项目（项目编号：12426306）及广东省基础与应用基础研究项目（项目编号：2023B0303000001）资助。Liyao Xiang的工作获国家自然科学基金委员会项目（项目编号：U25A20445和62272306）资助。本文的早期版本部分曾发表于IEEE SPAWC 2025 [1]。源代码见https://github.com/zhujhz/DeepFP.git。（通讯作者：Kaiming Shen。）

Jianhang Zhu和Kaiming Shen现就职于香港中文大学（深圳）理工学院，地址：中国深圳市，邮编518172（电子邮箱：jianhangzhu1@link.cuhk.edu.cn；shenkaiming@cuhk.edu.cn）。Tsung-Hui Chang现就职于香港中文大学（深圳）人工智能学院，地址：中国深圳市，邮编518172（电子邮箱：tsunghui.chang@ieee.org）。Liyao Xiang现就职于上海交通大学，地址：中国上海市，邮编200240，China (e-mail: xiangliyao08@sjtu.edu.cn).

乘法；（ii）迭代算法需要为功率约束寻找最优拉格朗日乘子，这一过程复杂且高度非线性，并且会显著增加训练成本。为解决这些问题，本文提出了一种名为DeepFP的新型深度展开方案，该方案依托全新的FP技术。不同于以往方法[6]–[13]，DeepFP无需学习大矩阵求逆和拉格朗日乘子的非线性优化，而只需关注如何协调少量标量步长。下面概述本文的研究思路。波束成形问题的原始目标函数为 f_o，难以直接处理。传统FP方法 [4], [5] 建议将 f_o 转换为 f_q ，从而易于进行迭代优化，但由此带来的新问题是需要计算大矩阵逆。为消除这一复杂性，一种较新的方法——非齐次二次变换 [14] 进一步将 f_q转换为 f_n，但其性能对步长的选择十分敏感：如果步长过大，迭代可能无法收敛；如果步长过小，收敛速度则会变慢。为确定步长，[14] 提出了一个强假设，即同一小区中的所有用户使用相同的步长，并证明步长的上界由某个特征值决定。相比之下，本文考虑在整个网络中为每个单个用户分别调节步长，因此这一基于特征值的上界不再存在，步长的选择也可以更加激进。

WSR问题一直以来都非常难以求解。事实上，已有研究表明，即使对于单输入单输出情形，该问题也是NP难的 [15]。除[16], [17],中的分支定界方法外，现有大多数工作都旨在高效地寻找局部最优解。经典方法包括最大比传输（MRT） [18], 、迫零（ZF）方法 [19],以及正则化迫零预编码（RZF）方法 [20], ；这些方法在特定条件下已在链路级得到验证，但在系统级可能导致相当大的性能损失。在较新的文献中，加权最小均方误差算法 [2], [3] 被广泛用于求解WSR问题。其主要思想是利用速率最大化与均方误差（MSE）最小化之间的联系，将WSR问题重写为加权均方误差最小化问题——该问题可以通过块坐标下降（BCD）方法 [21] 以迭代方式高效求解。得益于BCD理论，加权最小均方误差算法能够保证收敛到WSR问题的驻点解。

然而，在多小区 MIMO 场景中，加权最小均方误差算法会带来较高的计算开销。具体而言，

表一 各发射机具有 N 根天线时 MIMO 波束成形的不同方法比较。

方法	能否避免 $N \times N$ 矩阵求逆?	能否避免调节拉格朗日乘子?	能否适用于多个小区?
加权最小均方误差 [2], [3]	✗	✗	✓
简化WMMSE [22]	（但可以缩小矩阵尺寸）	✓	✗
分式规划 [4]	✗	✗	✓
深度展开 + 分式规划 [6]	✗	✓	✓
FastFP [14], [23]	√（但需要进行特征值计算）	✓	✓
所提出的 DeepFP	✓	✓	✓

加权最小均方误差算法的每次迭代都需要对一个矩阵求逆，其大小与发射天线数量成正比。因此，在发射机侧部署大量天线的多小区 MIMO 场景中，该算法需要进行大量大型矩阵求逆。近期的一项工作 [22] 在一定程度上缓解了这一开销。 [22] 的核心思想是将波束成形向量映射到一个新的空间中，其维度仅取决于接收天线总数（或者用户数，假设每个用户只有一根接收天线）。这种改进后的加权最小均方误差算法（在 [22] 中称为简化加权最小均方误差）所需求逆的矩阵大小与用户数成正比。显然，只有当网络中的用户数量有限时，简化加权最小均方误差算法才能降低复杂度。

WMMSE 算法在多小区 MIMO 中面临的另一项挑战源于功率约束。具体而言，在每次迭代中，WMMSE 都需要为每个小区确定一个拉格朗日乘子，以满足波束成形向量的功率约束。最优拉格朗日乘子没有闭式解，通常通过二分搜索来求解 [2]。近期提出的“简化WMMSE”算法 [22] 已在一定程度上解决了这一问题。[22] 的作者证明，在单小区场景下，同时缩放所有波束成形向量以满足功率约束是最优的。然而，将上述结果推广至多小区场景并不容易。解决多小区 MIMO 波束成形问题的另一种方法基于流形优化 [24]。其主要思想是将波束成形变量限制在由功率约束定义的黎曼流形上，从而将约束优化转化为无约束优化。然而，流形方法只能在固定功率水平下优化波束成形向量，而 WMMSE 可以联合优化波束成形向量和功率。此外，该方法的性能仅在单小区网络中得到验证。

除了上述模型驱动方法外，针对多小区 MIMO 波束成形问题的数据驱动方法也引发了大量研究兴趣。与纯粹的黑箱学习方法 [25]–[28]不同，后者试图利用深度神经网络（DNN）的通用逼近能力来模仿现有优化方法（例如加权最小均方误差算法），而深度展开方法 [29]–[31] 则考虑了传统模型驱动算法的迭代结构，旨在学习每次迭代的行为。对于加权和速率波束成形问题，先前的研究 [6]–[13]大多将加权最小均方误差算法作为深度展开的学习目标。例如，深度展开网络 [6] 针对简化加权最小均方误差算法，而 [7], [8] 针对加权最小均方误差算法。然而，这些深度展开方法在成功模仿加权最小均方误差算法的同时，也继承了其目标算法的上述缺点。因此，[6] 中的深度展开网络只能处理单小区场景， [8] 必须以次优的近似方式处理大型矩阵求逆，而 [7] 则局限于多输入单输出（MISO）场景，以避免学习如何通过二分搜索求解最优拉格朗日乘子。

为克服上述瓶颈，本文提出的深度展开方法利用了加权最小均方误差与分式规划之间的密切联系。粗略地说，分式规划指一类具有分式结构的优化问题，例如比值和最大化。事实证明，WSR问题可以改写为比值和问题，因此，加权最小均方误差算法归结为FP算法的一种特殊情况 [4], [5]。事实上，FP 领域已对大型矩阵求逆和拉格朗日乘子优化进行了充分研究，例如，所谓的非齐次二次变换 [14], [23] 能够同时解决这两个问题。因此，不同于以往直接对加权最小均方误差算法进行深度展开的工作[6]–[13] ，本文提出将非齐次二次变换融入深度展开范式。随后我们指出，学习任务的核心是确定基于非齐次二次变换的FP所使用的步长。本文方法的主要特点和优势总结于表一。我们的工作有两项主要创新：

- 深度展开的新范式：现有的波束成形深度展开方法通常模仿传统分式规划（基于二次变换）或加权最小均方误差算法的行为。相比之下，本研究基于非齐次二次变换，该变换本身无需计算复杂的大矩阵逆。此外，不同于 [14] 中逐小区调整步长参数的现有非齐次二次变换方法，本研究创新性地提出针对每个单个用户优化步长参数。数值结果表明，这种新范式能够实现更快的收敛和更优的性能。

- 一种新的混合训练策略：所提出的 DeepFP 采用一种结合监督学习与无监督学习的新型混合训练策略。具体而言，在第一阶段，利用模型驱动的 FastFP 解作为标签，对深度神经网络进行初始化和训练，以确保快速

$$f_q(\mathrm{V}, \underline{\boldsymbol{\Gamma}}, \mathrm{Y}) = \sum_k \Big[\mathrm{tr}\left(2\Re\left\{\mathrm{V}_{\ell k}^H \boldsymbol{\Lambda}_{\ell k}\right\} - \omega_{\ell k} \mathrm{Y}_{\ell k}^H \mathrm{D}_{\ell k} \mathrm{Y}_{\ell k} (\mathrm{I}_d + \boldsymbol{\Gamma}_{\ell k})\right) + \omega_{\ell k} \log |\mathrm{I}_d + \boldsymbol{\Gamma}_{\ell k}| - \mathrm{tr}\left(\omega_{\ell k} \boldsymbol{\Gamma}_{\ell k}\right) \Big]_{\ell}, \tag{7}$$

收敛并实现稳健的初始化。随后，在第二阶段，使用实际的加权和速率目标作为损失函数对深度神经网络进行微调，使网络能够针对真实目标进一步优化性能。

本文其余部分安排如下。第二节介绍加权和速率问题的建模。第三节展示并比较现有的模型驱动算法，包括FP算法和 FastFP算法。第四节基于 FastFP算法构建我们提出的 DeepFP 网络。第五节给出数值结果。最后，第六节总结全文。

本文中，粗体小写字母表示向量，粗体大写字母表示矩阵。对于向量 a，a^H 表示其共轭转置。对于矩阵 A，A^H表示其共轭转置，$\|\mathrm{A}\|_F$ 表示其弗罗贝尼乌斯范数。col(A)表示矩阵 A中的列数。对于方阵 A，tr(A)表示其迹，|A| 表示其行列式，$\lambda_{\max}$(A)表示其最大特征值。用 I_d 表示 $d \times d$单位矩阵，$\mathbb{C}^n$ 表示 $n \times 1$ 向量集合，$\mathbb{C}^{d\times n}$ 表示$d \times n$ 矩阵集合，$\mathbb{H}_+^{d\times d}$表示 $d \times d$ 正定矩阵集合。对于复数 $\mathrm{a} \in \mathbb{C}$，$\Re\{\mathrm{a}\}$ 表示其实部，|a| 表示其绝对值。带下划线的字母表示相关向量或矩阵的集合，例如，对于 a_1，.....，$\mathrm{a}_n \in \mathbb{C}^d$，我们记作 $\underline{\mathrm{a}} = [\mathrm{a}_1, \mathrm{a}_2, \ldots\ldots, \mathrm{a}_n]^\top \in \mathbb{C}^{n\times d}$。

II. 加权和速率最大化问题 LEM

考虑一个包含 L 个小区的下行多用户多输入多输出（MU-MIMO）系统。每个小区内有一个配备 N_t 根发射天线的基站（BS），为 K 个用户提供服务。第 ℓ个小区中的第 k 个用户记为（ℓ, k）。假设用户（ℓ, k）配备 N_r 根接收天线，且向其传输 d 个数据流。令 $\mathrm{V}_{\ell k} \in \mathbb{C}^{N_t\times d}$ 表示基站 ℓ 用于向用户（ℓ, k）传输信号 $\mathrm{s}_{\ell k} \in \mathbb{C}^{d\times 1}$ 的波束成形矩阵。假设 $\mathbb{E}[\mathrm{s}_{\ell k}\mathrm{s}_{\ell k}^H] = \mathrm{I}_d$，则用户（$\ell, k$）接收到的信号$y_{\ell k}$ 为

$$\mathbf{y}_{\ell k} = \underbrace{\mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell k}\mathbf{s}_{\ell k}}_{\text{desired signal}} + \underbrace{\sum_{j=1, j\neq k}^{K} \mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell j}\mathbf{s}_{\ell j}}_{\text{intracell interference}} + \underbrace{\sum_{i=1, i\neq \ell}^{L}\sum_{j=1}^{K} \mathbf{H}_{\ell k,i}\mathbf{V}_{ij}\mathbf{s}_{ij}}_{\text{intercell interference}} + \mathbf{n}_{\ell k}, \tag{1}$$

其中，信道状态信息（CSI）$\mathrm{H}_{\ell k,\ i} \in \mathbb{C}^{N_r\times N_t}$ 表示从基站 i到用户（ℓ, k）的信道，而 $\mathrm{n}_{\ell k} \sim \mathcal{CN}(0, \sigma^2\mathrm{I})$ 表示功率为 σ^2 的加性高斯白噪声。用户（ℓ, k）的可达数据速率可计算为 [32]

$$R_{\ell k} = \log|\mathbf{I} + \mathbf{V}_{\ell k}^H\mathbf{H}_{\ell k,\ell}^H\mathbf{F}_{\ell k}^{-1}\mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell k}|, \tag{2}$$

算法1：用于多小区 MIMO 波束成形的分式规划

1：输入：当前信道状态信息。
2：在功率约束下，将 V 初始化为可行值。
3: 重复
4: 根据（9）更新每个 $\mathrm{Y}_{\ell k}$。
5: 根据（10）更新每个 $\boldsymbol{\Gamma}_{\ell k}$。
6: 根据（11）更新每个 $\mathrm{V}_{\ell k}$。
7：直到目标值收敛
8：输出：最终波束成形矩阵 V

其中

$$\mathbf{F}_{\ell k} = \sum_{j=1, j\neq k}^{K} \mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell j}\mathbf{V}_{\ell j}^H\mathbf{H}_{\ell k,\ell}^H + \sum_{i=1, i\neq \ell}^{L}\sum_{j=1}^{K} \mathbf{H}_{\ell k,i}\mathbf{V}_{ij}\mathbf{V}_{ij}^H\mathbf{H}_{\ell k,i}^H + \sigma^2\mathbf{I}_{N_r}. \tag{3}$$

我们寻求最优发射波束成形器 V，以最大化加权和速率:

$$\max_{\underline{\mathbf{V}}} \quad f_o(\underline{\mathbf{V}}) := \sum_{\ell=1}^{L}\sum_{k=1}^{K} w_{\ell k} R_{\ell k} \tag{4a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \mathrm{tr}(\mathbf{V}_{\ell k}\mathbf{V}_{\ell k}^H) \leq P_\ell,\ \ell = 1, 2, \ldots, L, \tag{4b}$$

其中，非负权重 $w_{\ell k} \geq 0$ 反映了用户（ℓ、k）的优先级，常数 P_ℓ 是基站 ℓ 的功率预算。

III. 现有的基于优化的方法

A. FP Method

通过拉格朗日对偶变换 [5], ，将原始目标函数f_o(V)转换为

$$f_r(\underline{\mathbf{V}}, \underline{\boldsymbol{\Gamma}}) = \sum_{\ell=1}^{L}\sum_{k=1}^{K} w_{\ell k}\left[\log|\mathbf{I}_d + \boldsymbol{\Gamma}_{\ell k}| - \mathrm{tr}(\boldsymbol{\Gamma}_{\ell k}) + \mathrm{tr}((\mathbf{I} + \boldsymbol{\Gamma}_{\ell k})\mathbf{V}_{\ell k}^H\mathbf{H}_{\ell k,\ell}^H\mathbf{D}_{\ell k}^{-1}\mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell k})\right], \tag{5}$$

其中

$$\mathbf{D}_{\ell k} = \sum_{i=1}^{L}\sum_{j=1}^{K} \mathbf{H}_{\ell k,i}\mathbf{V}_{ij}\mathbf{V}_{ij}^H\mathbf{H}_{\ell k,i}^H + \sigma^2\mathbf{I}_{N_r}. \tag{6}$$

随后，FP方法应用二次变换 [4] ，进一步将 f_o(V) 改写为 f_q(V, $\underline{\boldsymbol{\Gamma}}$,Y)，如式（7）所示，其中

$$\boldsymbol{\Lambda}_{\ell k} = w_{\ell k}\mathbf{H}_{\ell k,\ell}^H\mathbf{Y}_{\ell k}(\mathbf{I}_d + \boldsymbol{\Gamma}_{\ell k}). \tag{8}$$

新的目标函数 f_q(V,$\underline{\boldsymbol{\Gamma}}$,Y) 关于 V $\underline{\boldsymbol{\Gamma}}$ 和 Y 分别是凹函数，因此 FP算法可以迭代优化这些变量，如下所示:

$$\mathbf{Y}_{\ell k} = \mathbf{D}_{\ell k}^{-1}\mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell k}, \tag{9}$$

$$\boldsymbol{\Gamma}_{\ell k} = \mathbf{V}_{\ell k}^H\mathbf{H}_{\ell k,\ell}^H\mathbf{F}_{\ell k}^{-1}\mathbf{H}_{\ell k,\ell}\mathbf{V}_{\ell k}, \tag{10}$$

$$\mathbf{V}_{\ell k} = (\eta_\ell\mathbf{I}_{N_t} + \mathbf{L}_\ell)^{-1}\boldsymbol{\Lambda}_{\ell k}, \tag{11}$$

$$f_n(\underline{\mathbf{V}},\underline{\mathbf{\Gamma}},\underline{\mathbf{Y}},\underline{\mathbf{Z}}) = \sum_{\ell,k}\Big[\operatorname{tr}\left(2\Re\left\{\mathbf{V}_{\ell k}^H\mathbf{\Lambda}_{\ell k}+\mathbf{V}_{\ell k}^H(\lambda_\ell\mathbf{I}_{N_t}-\mathbf{L}_\ell)\mathbf{Z}_{\ell k}\right\}+\mathbf{Z}_{\ell k}^H(\mathbf{L}_\ell-\lambda_\ell\mathbf{I}_{N_t})\mathbf{Z}_{\ell k}-\lambda_\ell\mathbf{V}_{\ell k}^H\mathbf{V}_{\ell k}\right)$$
$$-\operatorname{tr}\left(\omega_{\ell k}\sigma^2(\mathbf{I}_d+\mathbf{\Gamma}_{\ell k})\mathbf{Y}_{\ell k}^H\mathbf{Y}_{\ell k}\right)+\omega_{\ell k}\log|\mathbf{I}_d+\mathbf{\Gamma}_{\ell k}|-\operatorname{tr}\left(\omega_{\ell k}\mathbf{\Gamma}_{\ell k}\right)\Big] \quad (15)$$

其中

$$\mathbf{L}_\ell=\sum_{i=1}^{L}\sum_{j=1}^{K}w_{ij}\mathbf{H}_{ij,\ell}^H\mathbf{Y}_{ij}(\mathbf{I}_d+\mathbf{\Gamma}_{ij})\mathbf{Y}_{ij}^H\mathbf{H}_{ij,\ell}, \quad (12)$$

功率约束对应的拉格朗日乘子 η_ℓ （见式（11））计算如下:

$$\eta_\ell=\min\left\{\eta\ge 0:\sum_{k=1}^{K}\operatorname{tr}(\mathbf{V}_{\ell k}\mathbf{V}_{\ell k}^H)\le P_\ell\right\}, \quad (13)$$

如算法1所示。注意，WMMSE算法 [2], [3]是算法1的一个特例。

B. FastFP方法 *[14], [23]*

FP方法的主要缺点是需要计算式（11）中的大矩阵逆：回想一下，在多小区 MIMO 场景中，$\mathbf{L}_\ell$ 是一个 $N_t\times N_t$ 矩阵，而 N_t 是一个很大的数。为避免大矩阵求逆，我们可以将以下界引入 FP 方法:

引理 1.（非齐次界 *[33]*）设两个厄米特矩阵 $\mathbf{L},\mathbf{K}\in\mathbb{H}^{m\times m}$ 满足 $\mathbf{L}\preceq\mathbf{K}$。则对于任意两个矩阵 $\mathbf{X},\mathbf{Z}\in\mathbb{C}^{m\times m}$，有

$$\operatorname{tr}(\mathbf{X}^H\mathbf{L}\mathbf{X})\le\operatorname{tr}\left(\mathbf{X}^H\mathbf{K}\mathbf{X}+2\Re\{\mathbf{X}^H(\mathbf{L}-\mathbf{K})\mathbf{Z}\}+\mathbf{Z}^H(\mathbf{K}-\mathbf{L})\mathbf{Z}\right), \quad (14)$$

其中，当 $\mathbf{Z}=\mathbf{X}$ 时等号成立。

备注 1。简而言之，我们希望在式 (14) 中选取一个易于求逆的矩阵 L，因此自然可以令 $\mathbf{L}=\lambda\mathbf{I}$。接下来需要选择 $\lambda\in\mathbb{R}$，使其满足条件 $\mathbf{L}\preceq\mathbf{K}$。我们可以计算 K 的最大特征值，并令 $\lambda=\lambda_{\max}(\mathbf{L})$。另一种做法是令 $\lambda=\|\mathbf{L}\|_F$，但这样会增大 L 与 K 之间的差距，导致收敛速度变慢。

根据引理 1 和备注 1，我们进一步将f_q(V, $\underline{\mathbf{\Gamma}}$, Y)改写为 f_n (V, $\underline{\mathbf{\Gamma}}$, Y, Z)，如式 (15) 所示，其中

$$\lambda_\ell=\lambda_{\max}(\mathbf{L}_\ell) \quad (16)$$

固定其他变量时，式 (15) 中的每个 Z 均可最优地确定为

$$\mathbf{Z}_{\ell k}=\mathbf{V}_{\ell k}. \quad (17)$$

固定其他变量后，当前迭代 t 的每个 $\mathbf{V}_{\ell k}$ 均根据上一迭代 $t-1$ 的结果进行最优确定，如下所示

$$\mathbf{V}_{\ell k}^{(\tau)}=\begin{cases}\widehat{\mathbf{V}}_{\ell k} & \text{if } \sum_{j=1}^{K}\|\widehat{\mathbf{V}}_{\ell j}\|_F^2\le P_\ell\\ \sqrt{\frac{P_\ell}{\sum_{j=1}^{K}\|\widehat{\mathbf{V}}_{\ell j}\|_F^2}}\widehat{\mathbf{V}}_{\ell k} & \text{otherwise,}\end{cases} \quad (18)$$

算法2 用于多小区 MIMO 波束成形的 FastFP

1：输入：当前信道状态信息。2：在功率约束下将 V 初始化为可行值。3：重复执行 4：根据 (17) 更新每个 Z $_{\ell k}$。5：根据 (9) 更新每个 $\mathrm{Y}_{\ell k}$。6：根据 (10) 更新每个 $\mathbf{\Gamma}_{\ell k}$。7：根据 (18) 更新每个 $\mathrm{V}_{\ell k}$。8：直到目标值收敛 9：输出：最终波束成形矩阵 V

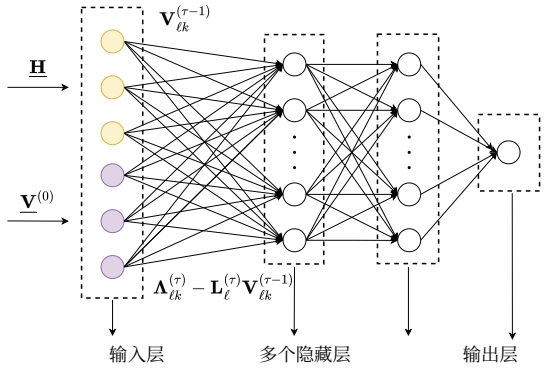


图 1. DeepFP 网络中使用的深度神经网络结构。该深度神经网络由一个输入层、多个隐藏层和一个输出层组成。隐藏层中的激活函数是 ReLU 的复数扩展。

其中

$$\widehat{\mathbf{V}}_{\ell k}=\mathbf{V}_{\ell k}^{(\tau-1)}+\frac{1}{\lambda_\ell}(\mathbf{\Lambda}_{\ell k}-\mathbf{L}_\ell\mathbf{V}_{\ell k}^{(\tau-1)}). \quad (19)$$

在本文中，我们使用上标 τ 或 $\tau-1$ 来标记迭代。$\mathrm{Y}_{\ell k}$ 和 $\mathbf{\Gamma}_{\ell k}$ 的最优更新与式 (9) 和 (10) 中相同。算法 2 总结了 FastFP方法。

IV. 所提出的 DeepFP 用于多小区 MIMO 波束成形

回顾一下，FastFP方法需要计算 L 的最大特征值，以确定每个 λ_ℓ，因此会产生三次计算复杂度。本节介绍一种称为 DeepFP 的深度展开方法，该方法无需显式特征值计算即可选择 λ_ℓ。

A. 面向迭代优化的深度展开

通用迭代算法可以写成如下标准形式： [6]

$$\underline{\mathbf{x}}^{(\tau)}=f_\tau(\underline{\mathbf{x}}^{(\tau-1)};\underline{\phi}), \quad (20)$$

其中，$\tau=1$、2、.....、T表示迭代索引，T为总迭代次数，x 为优化变量，

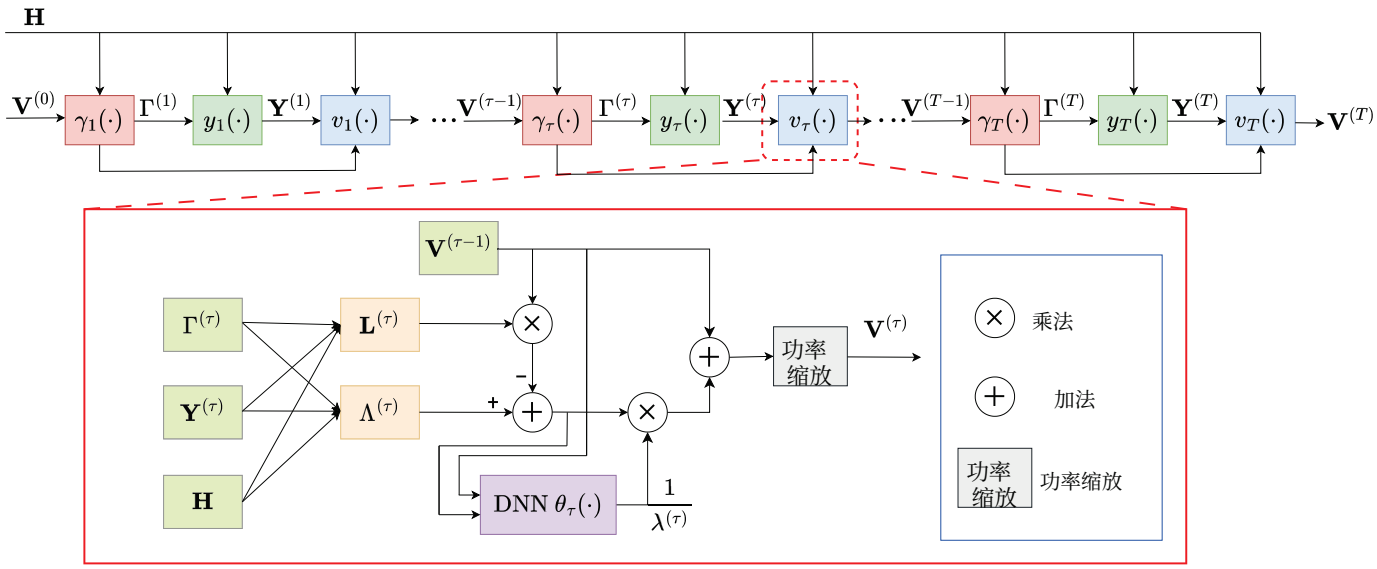


图 2. 我们所提出的 DeepFP 网络架构。模块 $z_\tau(\cdot)$ $\gamma_\tau(\cdot)$ $y_\tau(\cdot)$ $v_\tau(\cdot)$ 分别基于 (17)、(10)、(9) 和 (18) 设计。在模块 $v_\tau(\cdot)$ 中，参数 $\lambda^{(\tau)}$ 由深度神经网络 θ_τ 提供，$\mathrm{L}^{(\tau)}$ 由 (12) 确定，而 $\mathbf{\Lambda}^{(\tau)}$ 由 (8) 确定。DeepFP 网络不同层中的深度神经网络结构相同，但参数不共享。

状态变量 $\underline{\phi}$ 是一个随机变量，用于表征优化问题中的不确定性（例如，它是 MIMO 波束成形问题中的随机信道衰落），而 f_τ 是迭代函数，它根据当前状态 $\underline{\phi}$，以先前的解 $\mathbf{x}^{(\tau)}$ 为条件，得到新的解 $\mathbf{x}^{(\tau-1)}$。

深度展开旨在将迭代算法展开为多层顺序过程。给定一组可训练参数 θ，深度展开方法将 (20) 表示为一个深度神经网络层：

$$\underline{\mathbf{x}}^{(\tau)} = \mathcal{F}_\tau(\underline{\mathbf{x}}^{(\tau-1)}; \theta_\tau, \underline{\phi}), \tag{21}$$

其中，$\tau = 1、2、\ldots、T$ 重复用作层索引，$\mathcal{F}_\tau$表示第 τ 层深度展开网络的结构，$\mathbf{x}^{(\tau-1)}$ 和 $\mathbf{x}^{(\tau)}$ 分别是第 τ 层的输入和输出。原则上，在 θ_τ 经过适当训练后，对于任意可能的 $\underline{\phi}$，$\mathcal{F}_\tau(\mathbf{x}^{(\tau-1)}; \theta_\tau, \underline{\phi})$ 预期会表现得与 $f_\tau(\mathbf{x}^{(\tau-1)}; \underline{\phi})$ 类似。

B. 通过深度神经网络优化 λ_ℓ

将上述深度展开框架专门应用于波束成形问题 (4) 和 FastFP算法，可得到以下对应关系：

$$\underline{\mathbf{x}} = \{\mathbf{\Gamma}_{\ell k}, \mathbf{Y}_{\ell k}, \mathbf{V}_{\ell k}\}, \tag{22}$$

$$\underline{\phi} = \{\mathbf{H}_{\ell k,j}, w_{\ell k}, P_\ell, \sigma^2\}. \tag{23}$$

公式 (18) 表明，FastFP 中 $\mathrm{V}_{\ell k}$ 的更新遵循梯度投影形式，其中标量 $1/\lambda_\ell$ 充当小区 ℓ 中所有用户更新的步长。

我们将 $\lambda_{\ell k}^{(\tau)}$ 视为 $\mathrm{V}_{\ell k}^{(\tau-1)}$ 和项 $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$ 的函数。令 $\theta_\tau(\cdot)$ 表示展开网络中的第 τ 个深度神经网络层。那么，$\lambda_{\ell k}^{(\tau)}$ 的值为

$$\lambda_{\ell k}^{(\tau)} = \theta_\tau(\mathbf{V}_{\ell k}^{(\tau-1)}, \mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathbf{L}_\ell^{(\tau)} \mathbf{V}_{\ell k}^{(\tau-1)}). \tag{24}$$

如式 (24) 所示，我们将 $\lambda_{\ell k}^{(\tau)}$ 视为 $\mathrm{V}_{\ell k}^{(\tau-1)}$ 和 $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$ 的函数。上述设定的依据如下：在 FastFP算法中，波束成形矩阵 $\mathrm{V}_{\ell k}^{(\tau)}$ 更新为其上一迭代的取值与新的方向矩阵 $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$ 的线性组合。当迭代次数较少时，$\mathrm{V}_{\ell k}^{(\tau)}$ 与新方向相差较大。因此，$\lambda_{\ell k}^{(\tau)}$ 应较大，以加快收敛。随着迭代次数增加，$\mathrm{V}_{\ell k}^{(\tau)}$ 逐渐接近驻点，而 $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$ 逐渐接近零向量。在这种情况下，$\lambda_{\ell k}^{(\tau)}$ 应较小，以避免振荡。因此，我们将 $\lambda_{\ell k}^{(\tau)}$ 建模为 $\mathrm{V}_{\ell k}^{(\tau-1)}$ 和 $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$ 的函数。

在 FastFP算法中，$\lambda_{\ell k}^{(\tau)}$ 被设为 L_ℓ 的最大特征值，以确保收敛。相比之下，所提出的 DeepFP 网络不必要求 $\lambda_{\ell k}^{(\tau)}$ 满足 (14)。相反，我们的目标是通过深度神经网络寻找理想的 $\lambda_{\ell k}^{(\tau)}$，从而得到更好的 $\mathrm{V}_{\ell k}^{(\tau)}$。通过选择一个小于 (16) 中对应值的 $\lambda_{\ell k}^{(\tau)}$，即可实现这一目标。根据主化-最小化 (MM) [33] 理论，可以通过优化加权和速率的下界，即代理函数 $f_n(\mathrm{V}\underline{\Gamma}, \mathrm{Y}, \mathrm{Z})$，来提升加权和速率。较小的 $\lambda_{\ell k}^{(\tau)}$ 可能带来更紧的下界，从而加速迭代过程。

DeepFP 网络所采用的深度神经网络结构由一个输入层、多个隐藏层和一个输出层组成，如图 1 所示。深度神经网络的输入是展平后的 $\mathrm{V}_{\ell k}^{(\tau-1)}$ and $\mathbf{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathrm{V}_{\ell k}^{(\tau-1)}$。我们不分别处理实部和虚部，而是直接将展平后的复矩阵用作深度神经网络的整体输入。为此，我们扩展了修正线性单元（ReLU） [34] 激活函数，使其支持隐藏层中的复值数据。具体而言，复数ReLU 定义如下：

$$\mathrm{ReLU}_{\mathrm{Complex}}(a + bi) = \max(a, 0) + \max(b, 0)i, \tag{25}$$

算法3：用于多小区 MIMO 波束成形的 DeepFP

–训练阶段–
输入：随机生成的信道样本。
阶段 1：监督学习
运行算法2以获得解 $\mathbf{V}^*$。
使用 $\underline{\mathbf{V}}^*$ 作为标签，基于损失函数训练深度神经网络(30)中的函数。
阶段 2：无监督学习
基于损失函数微调深度神经网络参数 在 (31) 中。
输出：优化后的深度神经网络参数。

–测试阶段–
输入：当前的信道状态信息。
将 $\mathbf{V}^{(0)}$ 初始化为满足功率约束的可行值。
对于 $\tau = 1$ 到 T，执行
$\underline{\boldsymbol{\Gamma}}^{(\tau)} \leftarrow \gamma_\tau(\underline{\mathbf{V}}^{(\tau-1)};\ \underline{\phi})$
$\underline{\mathbf{Y}}^{(\tau)} \leftarrow y_\tau(\underline{\mathbf{V}}^{(\tau-1)};\ \underline{\phi})$
$\lambda_{\ell k}^{(\tau)} \leftarrow \theta_\tau(\mathbf{V}_{\ell k}^{(\tau-1)}, \boldsymbol{\Lambda}_{\ell k}^{(\tau)} - \mathrm{L}_\ell^{(\tau)} \mathbf{V}_{\ell k}^{(\tau-1)})$
$\underline{\mathbf{V}}^{(\tau)} \leftarrow v_\tau(\underline{\mathbf{V}}^{(\tau-1)}, \underline{\boldsymbol{\Gamma}}^{(\tau)}, \mathrm{Y}^{(\tau)};\ \underline{\lambda}^{(\tau)}, \underline{\phi})$
结束循环
输出：波束成形矩阵 $\mathbf{V}^{(T)}$。

其中，i 是虚数单位。我们用 $\Re(\cdot)$ 表示输出层中的激活函数，以确保深度神经网络的输出为实数。

C. 展开层

对于深度神经网络 $\theta_\tau(\cdot)$，DeepFP 网络中第 τ 层的结构可描述为

$$\underline{\boldsymbol{\Gamma}}^{(\tau)} = \gamma_\tau(\underline{\mathbf{V}}^{(\tau-1)}; \underline{\phi}), \tag{26}$$

$$\underline{\mathbf{Y}}^{(\tau)} = y_\tau(\underline{\mathbf{V}}^{(\tau-1)}; \underline{\phi}), \tag{27}$$

$$\lambda_{\ell k}^{(\tau)} = \theta_\tau(\mathbf{V}_{\ell k}^{(\tau-1)}, \boldsymbol{\Lambda}_{\ell k}^{(\tau)} - \mathbf{L}_\ell^{(\tau)} \mathbf{V}_{\ell k}^{(\tau-1)}), \tag{28}$$

$$\underline{\mathbf{V}}^{(\tau)} = v_\tau(\underline{\mathbf{V}}^{(\tau-1)}, \underline{\boldsymbol{\Gamma}}^{(\tau)}, \underline{\mathbf{Y}}^{(\tau)}; \underline{\lambda}^{(\tau)}, \underline{\phi}), \tag{29}$$

其中，(26)、(27) 和 (29) 分别对应迭代算法步骤 (10)、(9) 和 (18)。尽管模块 $\gamma_\tau(\cdot)$ 和 $y_\tau(\cdot)$ 涉及矩阵求逆运算，但这里的矩阵仅为 $N_r \times N_r$，其中 N_r 是每个用户终端的接收天线数。由于本文关注的是大规模 MIMO 网络，N_r 通常远小于每个基站的发射天线数 N_t。因此，我们不太关注 $N_r \times N_r$ 矩阵求逆，而只关注消除模块 $v_\tau(\cdot)$ 中的 $N_t \times N_t$ 矩阵求逆。

但是，如果 N_r 也很大呢？实际上，消除 $N_r \times N_r$ 矩阵求逆在概念上与消除 $N_t \times N_t$ 并无不同。如 [14]，中所讨论的，我们只需进一步将引理 1 中的非齐次界应用于 $\mathrm{Y}_{\ell k}$ 和 $\boldsymbol{\Gamma}_{\ell \mathbf{k}}$ 的更新，尽管数学符号会复杂得多。

DeepFP 网络的完整结构如图 2 所示。变量 $\mathrm{L}^{(\tau)}$ 和 $\underline{\boldsymbol{\Lambda}}^{(\tau)}$ 分别根据式 (12) 和式 (8) 计算。展开网络中不同层的深度神经网络采用相同的结构（例如隐藏层的数量、每层的神经元数量以及激活函数）。“功率缩放”模块表示对波束成形向量进行缩放，以满足功率约束。该模块对应式 (18)，旨在确保深度神经网络生成的波束成形矩阵满足发射功率约束。

实际上，本文旨在回答的核心问题是：坚持 MM 性质是否值得。确实，MM 性质能够保证收敛，但代价是计算量增加——我们必须反复计算大型 $N_t \times N_t$ 矩阵的逆。非齐次 FP 可以避免矩阵求逆，但随后 λ_ℓ 难以确定。文献 [14] 表明，当小区 ℓ中的所有用户都使用相同的 λ_ℓ 时，$\lambda_\ell = \lambda_{\max}(L_\ell)$ 可以保证 MM 性质；但这会牺牲性能，因为它限制了选择 λ_ℓ的灵活性。相比之下，本文放宽了 MM 约束，允许针对每个单个用户分别调整 λ_ℓ。

D. 训练策略

我们采用包含两个阶段的混合训练策略。第一阶段，对于给定的信道样本 H，令 $\mathbf{V}^*$ 表示通过 FastFP算法（算法2）获得的解，令 $\mathbf{V}^{(T)}$ 表示展开网络在最后一层的输出。第一训练阶段采用监督学习，其中 $\mathbf{V}^*$ 作为样本 H 对应的标签。第一阶段将 $\mathbf{V}^*$ 与 $\underline{\mathbf{V}}^{(T)}$ 之间的均方误差作为损失函数：

$$\mathrm{LOSS}_1 = \frac{1}{KL} \sum_{\ell=1}^{L} \sum_{k=1}^{K} \|\mathbf{V}_{\ell k}^{(T)} - \mathbf{V}_{\ell k}^*\|_2^2. \tag{30}$$

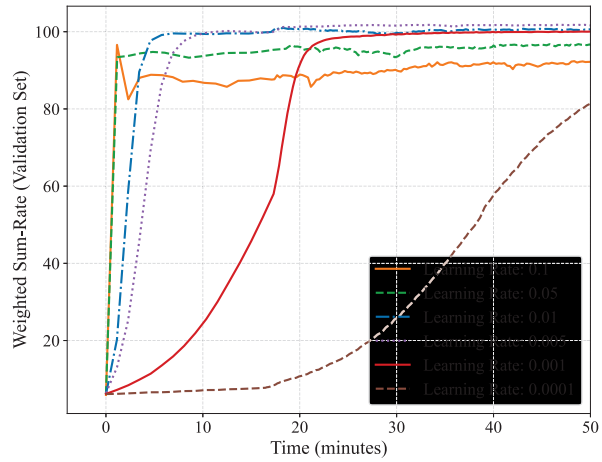
在第二阶段，我们转而采用无监督学习，将 $\mathbf{V}^{(T)}$ 的加权和速率函数作为损失函数，即

$$\mathrm{LOSS}_2 = -\frac{1}{KL} \sum_{\ell=1}^{L} \sum_{k=1}^{K} w_{\ell k} R_{\ell k}. \tag{31}$$

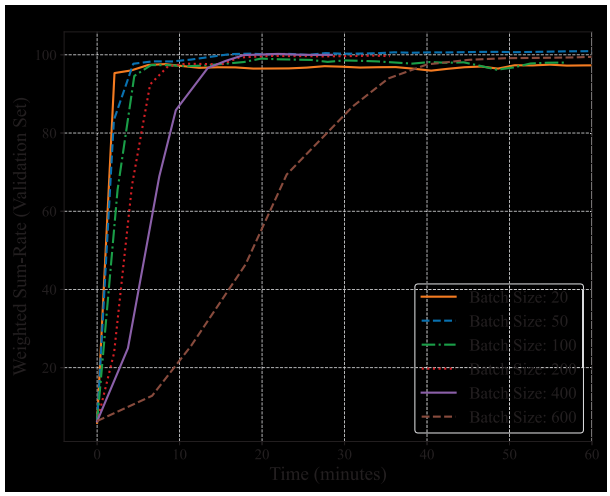
尽管已有研究表明完全无监督学习是可行的 [35]，但我们采用了这种结合监督学习和无监督学习的混合训练策略。

关于参数初始化，每个 $\mathbf{V}^{(0)}$ 均根据标准复高斯分布 $\mathcal{CN}(0, 1)$ 随机且独立地生成，随后进行缩放，以满足每个基站的功率约束 $\sum_{k=1}^{K} \mathrm{tr}(\mathbf{V}_{\ell k}\mathbf{V}_{\ell k}^H) = P_\ell$。此外，在第一阶段的监督学习中，FastFP算法和展开网络使用相同的初始点 $\mathbf{V}^{(0)}$。算法3总结了DeepFP方法的训练和推理流程。

在训练阶段，我们使用大量从同一分布中随机生成的信道状态信息来训练深度神经网络。训练完成后，深度神经网络便固定下来，可以视为一个确定性函数：以具体的信道状态信息值为输入，并输出相应结果。换言之，深度神经网络本身取决于信道分布，而非具体的信道状态信息值。因此，当针对来自同一分布的新信道状态信息实现考虑波束成形问题时，无需重新训练深度神经网络。



(a) 不同的学习率选择。



(b) 不同的批量大小选择。

图3. 不同学习率 (a) 和批量大小 (b) 下训练过程中验证数据集上的 WSR 性能。

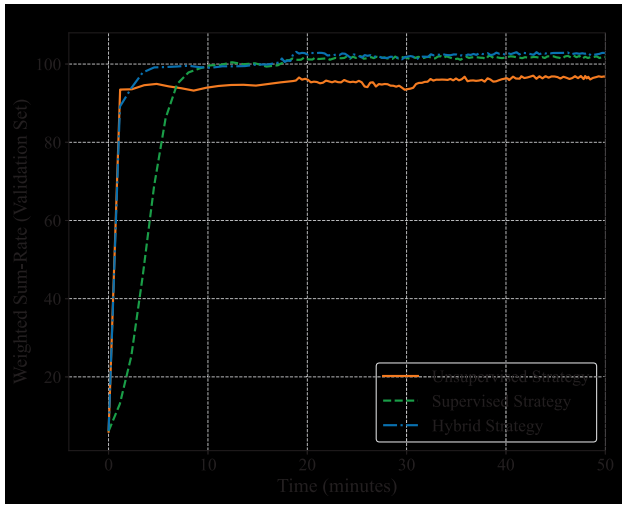


图 4. 三种训练策略的验证性能对比：无监督、监督和混合。曲线显示了训练期间的加权和速率（WSR），各策略呈现出不同的收敛行为。

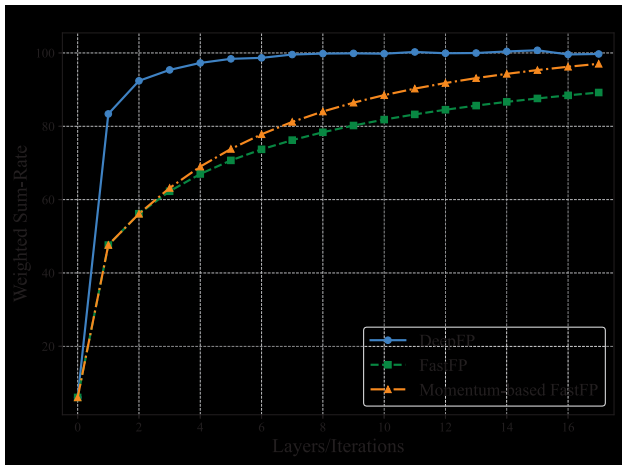


图 5. DeepFP 网络、FastFP算法和基于动量的FastFP算法的加权和速率性能。对于 FastFP 和基于动量的 FastFP，曲线表示经过 i 次迭代后的加权和速率（WSR）。对于 DeepFP，曲线显示了具有 i 层的网络所达到的加权和速率。

当前的 DeepFP 需要集中式训练，但有多种方法可以将其分布式化。一种可能的方法是将小区划分为若干簇，然后按簇训练深度神经网络。另一种可能的方法是在训练阶段引入联邦学习。

E. 复杂度分析

考虑一个 L 小区多输入多输出系统，其中每个小区配备 N_t 根发射天线，并为 K 个用户提供服务。每个用户配备 N_r 根接收天线。数据流数量为 d。算法2在每次迭代中的计算复杂度为

$$\mathcal{O}\left(LN_t^3 + L^2K^2(N_t+N_r)N_rd + LKN_r^2(N_r+d) + LK^2(N_t+d)d^2\right), \quad (32)$$

其中，项 $\mathcal{O}(LN_t^3)$ 用于计算式 (16) 中的最大特征值。

在 DeepFP 网络中，最大特征值计算由深度神经网络的前向传播取代，其他操作均保持不变。因此，每层的计算复杂度为

$$\mathcal{O}\left(LKU(2N_td + (M_{\text{hid}}-1)U + 1) + L^2K^2(N_t+N_r)N_rd + LKN_r^2(N_r+d) + LK^2(N_t+d)d^2\right), \quad (33)$$

其中，M_{hid} 表示深度神经网络中的隐藏层数，U 表示每个隐藏层中的神经元数量。与 FastFP算法相比，DeepFP 网络的计算复杂度更低。值得强调的是，基站的发射天线数量 N_t 是主要影响因素，尤其是在大规模 MIMO 网络中。对于 N_t，(32)表明，基于特征值的基准方法的复杂度为 $O(N_t^3)$；而(33)表明，我们的 DeepFP 方法的复杂度显著更低，为 $O(N_t)$。

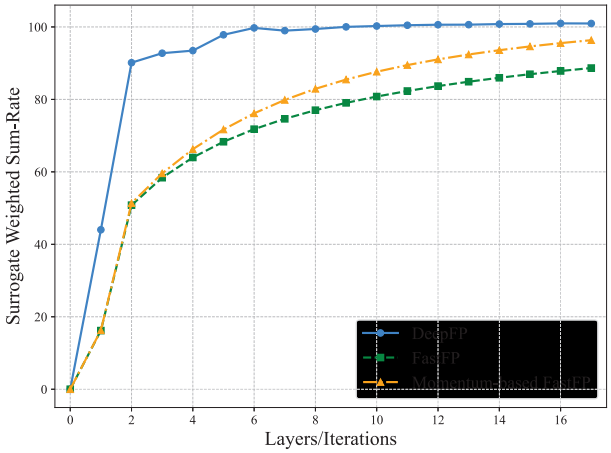


图 6. DeepFP 网络和 FastFP算法的代理加权和速率性能。

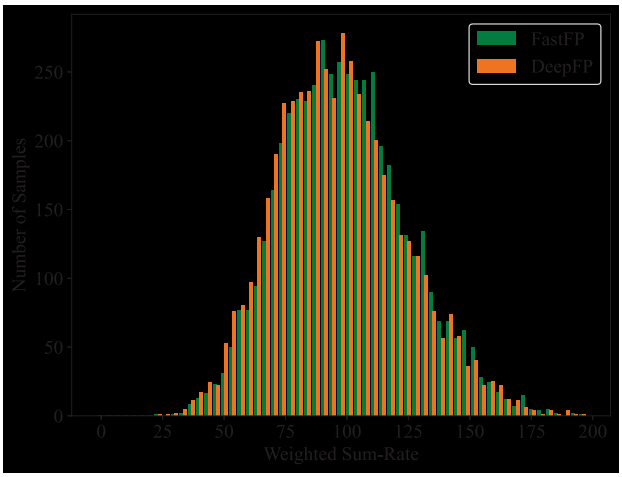


图 9. 在 $N_t = 64, N_r = 4, d = 2, K = 6$ 条件下，7 小区 MIMO 中 DeepFP 网络和 FastFP算法的分布。

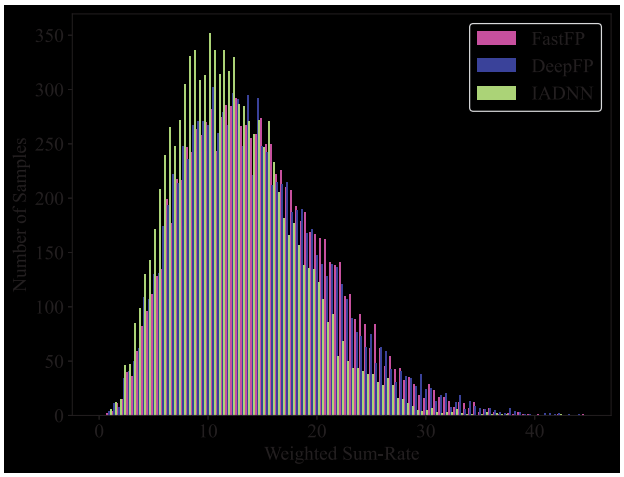


图 7. 在 $N_t = 64, N_r = 4, d = 2, K = 6$ 条件下，单小区 MIMO 系统中 DeepFP 网络和基线算法的分布。

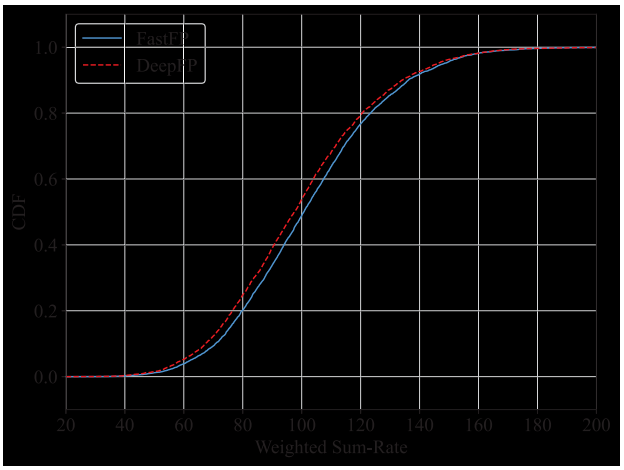


图 10. 在 $N_t = 64, N_r = 4, d = 2, K = 6$ 条件下，描述 7 小区 MIMO 中不同算法所实现速率的累积分布函数。

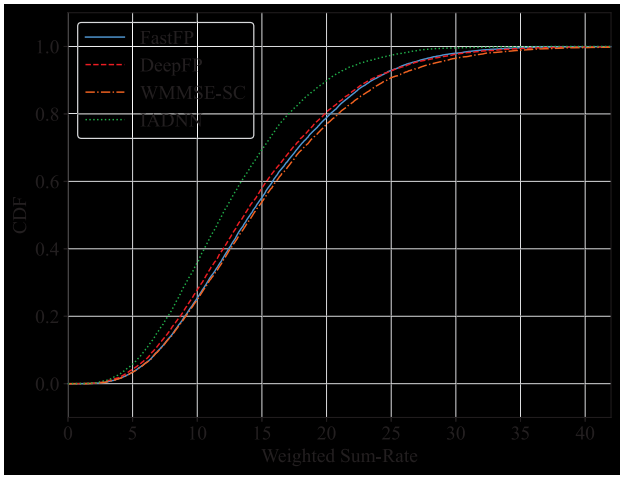


图 8. 在 $N_t = 64, N_r = 4, d = 2, K = 6$ 条件下，描述单小区 MIMO 系统中不同算法所实现速率的累积分布函数。

V. 实验结果

本节评估所提出的 DeepFP 网络相对于模型驱动算法和现有展开算法的性能。首先，我们评估不同训练策略对优化性能的影响。接着，我们测试多种无线网络示例。最后，我们通过采用不同的训练和测试设置，验证所提出的 DeepFP 网络的泛化能力。所提出的 DeepFP 网络使用 Python 3.10.0 和 PyTorch 2.4.1 实现。系统运行于一台台式机上，配备主频为 3.4 GHz 的 Intel i7-13700 中央处理器（CPU）和 64 GB 随机存取存储器（RAM）。训练时使用图形处理器（GPU）RTX 4080，以缩短训练时间；测试时则不使用。为确保一致性，包括 DeepFP 和经典基线方法在内的所有算法均使用 PyTorch 实现。为公平比较运行时间，所有推理测试均在 CPU 上进行，且不使用 GPU 加速。

表二 $N_t = 64$、$N_r = 4$、$d = 2$、$K = 6$下单小区多输入多输出的加权和速率与计算性能。

算法	加权和速率	CPU 时间（秒）
DeepFP	14.664 (98.9%)	0.053 (14.0%)
FastFP	14.826 (100.0%)	0.378 (100.0%)
FastFP（76 次迭代）	14.664 (98.9%)	0.287 (76.0%)
WMMSE-SC	15.270 (103.0%)	0.563 (148.9%)
IADNN	12.540 (84.9%)	0.055 (14.5%)

表 III 多小区 MIMO 在 $N_t = 64, N_r = 4, d = 2, K = 6$ 条件下的加权和速率和计算性能。

算法	加权和速率	CPU 时间（秒）
DeepFP	99.474（97.4%）	0.275（11.7%）
FastFP	102.186（100.0%）	2.333（100.0%）
FastFP（56 次迭代）	99.480 (97.4%)	1.306 (56.0%)
GCN-WMMSE	91.011 (89.1%)	0.604(25.9%)

表 IV 不同 K下多小区 MIMO 的 DeepFP 网络加权和速率与计算性能（$N_t = 64$，$N_r = 4$，$d = 4$）

K	加权和速率	CPU 时间（秒）	FastFP 迭代次数
6	128.796 (92.8%)	0.285 (11.2%)	23
9	157.554 (90.3%)	0.584 (12.1%)	21
15	203.895 (86.5%)	1.678 (7.6%)	19

表 V $N_t = 64, N_r = 4, K = 6$ 在不同 d 下的 DeepFP 网络加权和速率性能。DeepFP 网络使用 $d = 4$ 进行训练。

d	加权和速率（比特/秒）	FastFP 迭代次数
1	66.486（96.5%）	57
2	97.074（95.0%）	40
3	115.596（94.6%）	32
4	128.796（92.8%）	23

A. 实验设置

1) 数据集生成：我们根据 [14] 中的设定，从七小区六边形 MIMO 系统生成信道数据。在每个小区内，基站位于中心，下行用户随机分布。每个基站和用户分别配备 N_t 和 N_r 天线。数据流数量为 $d \leq N_r$。所有用户的权重均设为相等。相邻基站之间的距离为 $D = 0.8$ 千米。每个基站的最大发射功率为 20 dBm，背景噪声功率为 -90 dBm。下行链路的距离相关路径损耗建模为 $128.1+37.6\log_{10} r+\xi$（单位：dB），其中 r 表示基站与用户之间的距离（单位：千米）。 ξ 是一个均值为零、标准差为 8 dB 的高斯随机变量，用于描述阴影效应。

2）参数选择：在所有数值结果中，深度神经网络由两个隐藏层、一个输入层和一个输出层组成。除非另有说明，每个隐藏层包含 64 个神经元。我们首先研究批量大小和学习率对收敛性能的影响。我们设置$N_t = 64$、$N_r = 4$、$K = 6$和 $d = 2$。DeepFP 网络有 $T = 8$ 层。图 3 展示了在不同批量大小和学习率设置下，验证集的加权和速率在训练期间如何变化。结果表明，较大的学习率能够加快收敛。然而，学习率过大可能导致不稳定，并降低 WSR 性能。因此，根据图 3(a) 的结果，我们选择 0.005 作为初始学习率，并在训练过程中逐步降低。图 3(b) 的结果表明，随着批量大小增加，收敛速度先提高后降低。这是因为受内存限制，批量大小过大时，每个小批量的处理时间会更长。因此，我们选择 200 作为批量大小，以平衡 WSR 性能和收敛速度。我们进一步比较了三种训练策略：监督、无监督和混合。如图 4 所示，与单一训练方案相比，混合训练方案能够更好地权衡收敛速度和最终性能，

基准方法。

随后，我们分析 DeepFP 网络中的展开层数T 对 WSR 性能的影响。我们训练了展开层数各不相同的网络 T，并在测试集上评估其 WSR 性能。图 5 的结果表明，随着 T 增加，WSR 性能先提升，但当 T 超过 8 后开始波动。由于 DeepFP 网络的推理时间随 T线性增长，我们选择 $T = 8$ 以平衡 WSR 性能和推理时间。此外，图 5 展示了 DeepFP 网络相较于 FastFP算法和基于动量的 FP 的显著性能优势。结果表明，DeepFP 的性能优于基于 MM 性质的基准方法。当 DeepFP 网络的层数等于 FastFP算法的迭代次数时，DeepFP 网络的性能远优于 FastFP算法。我们还在图 6 中比较了 DeepFP、FastFP 和基于动量的 FastFP 在式（15）中的代理函数 $f_n(\cdot)$。如图所示，尽管 DeepFP 不再保证性能单调递增，但总体而言，偶尔出现的性能下降可以忽略不计；得益于更激进的步长选择，速率性能提升得快得多。

B. 不同无线网络的WSR最大化

1）单小区性能：我们评估 DeepFP 网络在不同网络规模下的加权和速率性能。首先考虑一个包含 $N_t = 64$、$N_r = 4$、 $K = 6$ 和 $d = 2$ 的单小区 MU-MIMO系统。我们选取以下三种算法作为基线算法：

1）FastFP算法：将算法2迭代100次后的结果作为 FastFP算法的输出。2）WMMSE-SC算法：WMMSE-SC算法首先使用加权最小均方误差求解一个无约束的加权和速率问题，然后缩放解以满足功率约束。该方法避免了二分法，但仍需进行大量矩阵求逆，而且仅在单小区情形下具有理论保证。

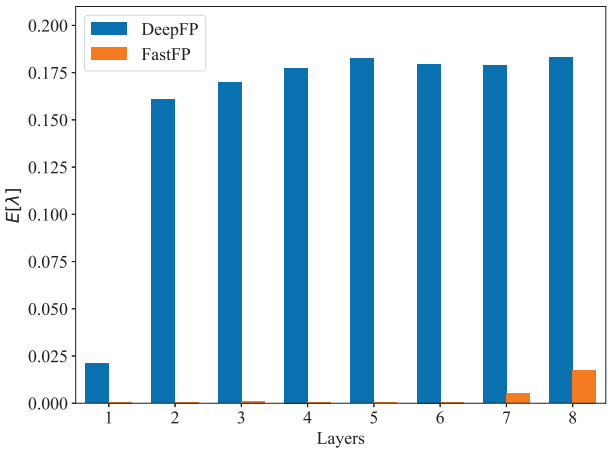


图 11. FastFP算法和DeepFP网络所选 λ 的均值。FastFP算法根据式 (16) 计算 λ，而DeepFP网络中的 λ 由深度神经网络确定。

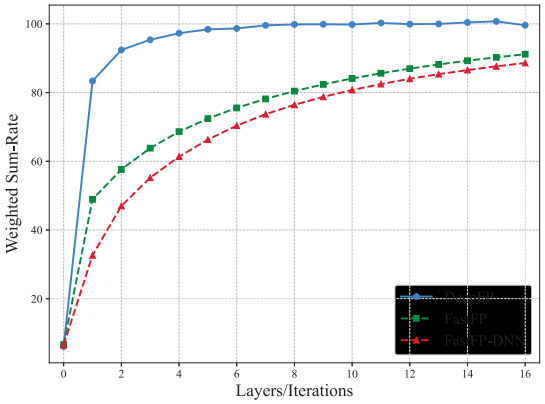


图 12. DeepFP 网络、FastFP算法和 FastFP-DNN算法的加权和速率性能。

表 VI 不同 N_t 下，DeepFP 网络使用$N_r = 4$，$d = 2$，$K = 6$ 时的加权和速率性能。DeepFP 网络使用 $N_t = 64$进行训练

N_t	加权和速率（比特/秒）	FastFP 迭代次数
16	55.224 (96.3%)	27
24	67.962 (96.7%)	39
32	77.502 (96.8%)	44
40	82.584 (96.5%)	42
48	86.880 (96.9%)	48
56	93.546 (96.8%)	49
64	99.474 (97.4%)	56

将迭代100次后的结果作为WMMSE-SC算法的输出。

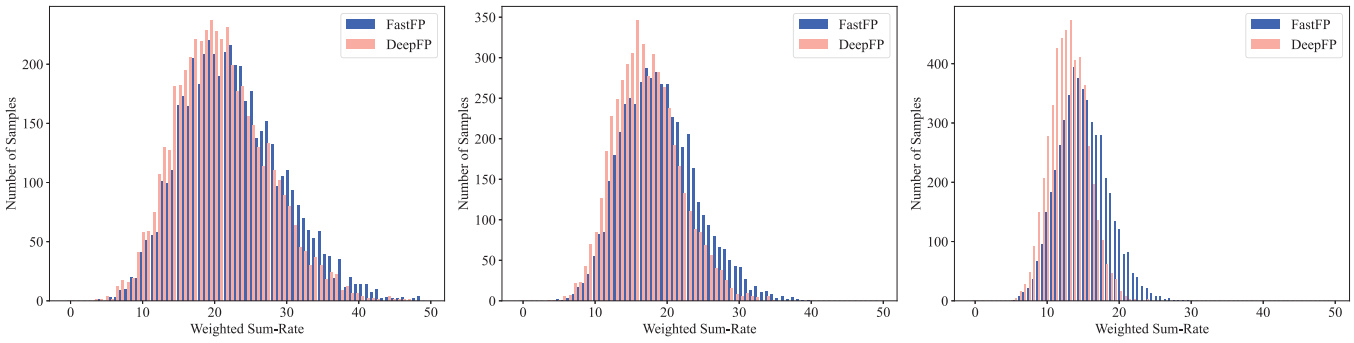
3) IADNN：迭代算法诱导的深度展开神经网络（IAIDNN） [6] 对单小区MIMO系统的加权最小均方误差-SC算法进行展开。IAIDNN通过引入可训练矩阵来消除大型矩阵求逆，这些矩阵基于一阶泰勒展开近似矩阵求逆。我们使用PyTorch实现了 [6] 中提出的原始网络结构，并遵循 [6] 中推荐的训练设置。与 [6] 中相同，IAIDNN的层数设为7。

我们使用相同的测试数据评估DeepFP网络和基线算法的加权和速率性能。根据10,000个测试样本计算平均加权和速率和CPU时间，结果见表 II。我们还报告了FastFP迭代76次后的结果，此时其加权和速率性能与DeepFP网络相同。为便于比较，我们以百分比形式将各算法的加权和速率性能和运行时间与FastFP算法进行对比。结果表明，DeepFP网络仅使用FastFP迭代100次所需运行时间的14.0%，便达到了其加权和速率的98.9%。FastFP算法需要迭代76次才能达到与DeepFP网络相同的加权和速率性能，运行时间因此接近其五倍。此外，我们的算法在计算时间相近的情况下，加权和速率性能优于IADNN。

图 7 和图 8 分别展示了不同算法所达到的 WSR 性能的分布及累积分布函数（CDF）。每种分布及其对应的累积分布函数均基于 10,000 个测试样本的结果。结果表明，所提出的 DeepFP 网络的性能与 FastFP算法非常接近，并且优于 IADNN 算法。

2）多小区性能：我们进一步验证了 DeepFP 网络在 7 小区环绕网络中的 WSR 性能。设置为 $N_t = 64$、$N_r = 4$、$K = 6$ 和 $d = 2$。FastFP 算法和 GCN-WMMSE 算法 [12] 作为两个基线。我们重新实现了 GCN-WMMSE 算法，并使用 PyTorch 调整其图神经网络的规模，使其与 DeepFP 中的深度神经网络规模相匹配，以便公平比较复杂度。表 III 展示了 WSR 性能和 CPU 推理运行时间。结果表明，所提出的 DeepFP 网络实现了 FastFP 所达到 WSR 的 97.4%，而运行时间仅为其 11.7 %。经过 56 次迭代后，FastFP 达到与 DeepFP 网络相同的性能。由于 GCN-WMMSE 旨在模仿传统加权最小均方误差算法的性能，因此它也继承了相关的弱点，例如收敛缓慢。与 GCN-WMMSE 相比，DeepFP 不仅取得了更高的 WSR 性能，而且推理延迟也更低。图 9 和图 10 展示了 WSR 的分布及其累积分布函数。结果表明，在多小区 MIMO 系统中，DeepFP 网络能够很好地近似 FastFP 的分布。图 11 展示了深度神经网络提供的 λ 均值，以及在相同输入下使用 (16) 计算得到的 λ 均值。结果符合预期：DeepFP 网络产生的 λ 值更小。

为增强实验的公平性，我们设计了一种新的基线算法，称为 FastFP-DNN。具体而言，我们使用深度神经网络直接预测 FastFP 中的最大特征值，并设计深度神经网络的规模，使其计算复杂度与 DeepFP 相同，从而对复杂度进行合理归一化。图 12 所示结果表明，对比算法的性能不如 FastFP，而 DeepFP 的性能则显著优于 FastFP。



(a) $K = 6$ (b) $K = 9$ (c) $K = 15$

图 13. 分布 在具有 $N_t = 64$、$N_r = 4$、$d = 4$ 的 7 小区 MIMO 中，DeepFP 网络和 FastFP算法在不同 K 下的分布。

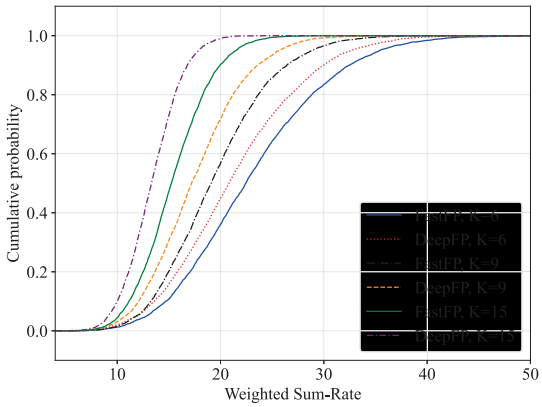


图 14. 描述在具有 $N_t = 64$、$N_r = 4$、$d = 4$ 的 7 小区 MIMO 中，DeepFP 网络和 FastFP算法在不同 K 下所达到速率的累积分布函数。

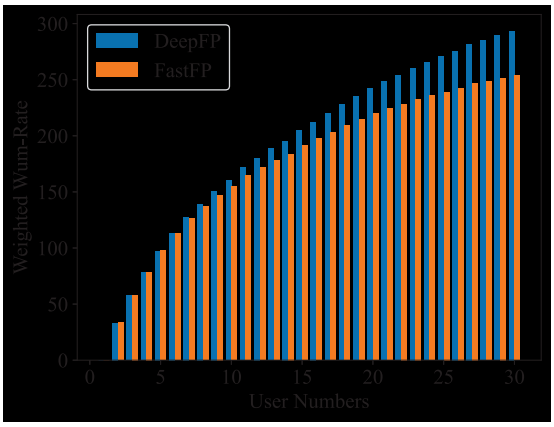


图 15. 在 7 小区 MIMO 中，当 $N_t = 64$、$N_r = 4$、$d = 2$ 时，不同 K 下 DeepFP 网络和 FastFP算法的加权和速率性能。DeepFP 网络使用 $N_t = 64$、$N_r = 4$、$d = 2$、$K = 6$ 进行训练。

接下来，我们考虑用户数量更多且每个用户的数据流数量更高的场景。我们设置 $N_t = 64$、$N_r = 4$、$d = 4$ 和$K = 6$、9、15。表 IV 给出了平均 WSR 性能和 CPU 时间。我们将“FastFP 迭代次数”定义为 FastFP 达到与 DeepFP 网络相同性能所需的平均迭代次数。结果表明，随着用户数量增加，WSR 性能有所提升。然而，DeepFP 网络与 FastFP 之间的差距也随之扩大。对比表 IV 和表 III，当 $N_t = 64$、$N_r = 4$ 和 $K = 6$ 时，DeepFP 网络在 $d = 2$ 下展现出更好的加速性能。图 13 和图 14 分别展示了不同 K 取值下 WSR 的分布和累积分布函数。结果表明，尽管随着用户数量增加，DeepFP 网络的 WSR 性能百分比有所下降，但它仍能较好地近似 FastFP 算法的分布。

C. 泛化能力验证

在上一小节中，我们评估了 DeepFP 网络在不同规模的 MIMO 系统中的加权和速率和加速性能。在实际应用中，测试数据往往与训练数据存在显著差异，这主要有两个原因：方面：1）测试数据的规模可能与训练数据不同。例如，在多小区 MIMO 中，用户数量可能因移动性而发生变化。此外，我们希望训练好的网络能够适用于来自不同小区的数据，因此发射天线的数量也会有所不同。2）数据分布发生变化。即使测试数据的规模与训练数据相同，其分布也可能不同。因此，在本小节中，我们评估 DeepFP 网络的泛化性能。

首先，我们使用通过 $N_t = 64$、$N_r = 4$、$K = 6$ 和 $d = 4$ 训练的网络，测试其在不同 d 取值下的性能。结果如表 V 所示。这些结果表明，对于不同的 d 取值，训练后的 DeepFP 网络在加权和速率方面仍表现良好。随着 d 减小，FastFP 达到相同性能所需的迭代次数增加。将 $d = 2$ 的结果与表 III 中的结果进行比较可以发现，每用户数据流数量增加时，网络的加权和速率性能下降了 2.4%。

接下来，我们测试使用 $N_t = 64$、$N_r = 4$、$K = 6$ 和 $d = 2$ 训练的 DeepFP 网络在 N_t 取不同值时的性能。结果如表 VI 所示。这些结果表明，随着 N_t 变化，DeepFP 网络的 WSR 性能仍稳定在约 97%。

接下来，我们继续使用经过训练的 DeepFP 网络

表 VII 不同小区距离下 DeepFP 网络的加权和速率性能 D 和标准差为 ξ：网络在 $D = 0.8$ 和标准差为 $\xi = 8$的条件下训练。

小区距离（千米）	加权和速率（比特/秒）			FastFP 迭代次数		
	ξ 4dB	ξ 8dB	ξ 12dB	ξ 4dB	ξ 8dB	ξ 12dB
0.4	261.168（93.2%）	261.012（101.5%）	259.386（113.2%）	58	132	233
0.6	167.178（94.8%）	186.222（99.1%）	205.962（109.5%）	45	91	197
0.8	108.090（96.1%）	136.746（99.2%）	162.756（107.5%）	35	89	152
1.0	73.386（96.5%）	100.506（99.1%）	127.536（105.2%）	25	82	217
1.2	52.062（97.0%）	76.446（98.8%）	105.948（103.5%）	21	71	204

表 VIII DeepFP 和 FastFP 在瑞利衰落信道中的和速率性能：网络在阴影模型下训练，并在无阴影的瑞利衰落环境中测试。

发射功率（dB）	加权和速率（比特/秒）	
	FastFP	DeepFP
0	56.7086	68.4735
10	62.5130	70.0218
20	66.9725	70.4233
30	71.3865	71.4206
40	76.2021	73.0454

$N_t = 64$、$N_r = 4$、$K = 6$和 $d = 2$，在用户数量各异的数据集上测试其 WSR 性能。结果如图 15 所示。这些结果表明，当$K < 6$时，经过 100 次迭代后，DeepFP 网络的表现优于 FastFP算法。随着 K 增大，DeepFP 网络与 FastFP算法之间的差距也随之扩大。

在之前的泛化测试中，我们评估了训练好的网络在不同规模测试数据上的泛化能力。接下来，我们测试该网络在不同分布数据上的泛化性能。在生成训练数据时，基站之间的距离设为$D = 0.8$ km，路径损耗参数 ξ 的标准差为 8 dB。我们生成具有不同 D 取值和标准差的测试数据，然后评估网络的 WSR 性能。结果如表 VII 所示。这些结果表明，DeepFP 网络在不同分布下均表现出良好的 WSR 性能。为进一步评估网络在不同信道统计特性下的泛化能力，我们在阴影衰落条件下训练网络，并在瑞利衰落场景中对其进行评估。如表 VIII 所示，在随机阴影衰落条件下训练的深度神经网络，在瑞利衰落模型中仍能获得可比的性能。

VI. 结论

本工作旨在提出一种新颖的深度展开范式，用于优化蜂窝网络中的多小区 MIMO 波束成形器。所提出的 DeepFP方法在两个方面区别于现有的用于 MIMO 波束成形的深度展开方法 [6]–[11]。首先，尽管以往的工作[22]只能降低大规模矩阵求逆的复杂度，但 DeepFP 完全消除了大规模矩阵求逆。其次，尽管以往的工作只能针对单小区对拉格朗日乘子优化进行线性化，但 DeepFP 将这种线性化扩展到了通用的多小区网络。DeepFP 通过将传统的加权最小均方误差算法 [2], [3] 与分式规划工具 [4], [5] 相结合，并进一步将非齐次界 [14] 纳入用于深度展开的深度神经网络设计中，获得了上述两项优势。大量数值示例表明，DeepFP 降低了传统模型驱动迭代算法（如加权最小均方误差）的复杂度，甚至能够在多小区 MIMO 网络中最大化加权和速率方面超越这些算法。

参考文献

[1] J. Zhu、T.-H. Chang、L. Xiang 和 K. Shen，“DeepFP：用于大规模 MIMO 波束成形的深度展开分式规划”，载于 IEEE 无线通信高级信号处理国际研讨会（SPAWC）论文集，2025年6月。[2] Q. Shi、M. Razaviyayn、Z.-Q. Luo 和 C. He，“针对 MIMO 干扰广播信道分布式和效用最大化的迭代加权最小均方误差方法”，IEEE《信号处理汇刊》，第59卷，第9期，第4331–4340页，2011年9月。[3] S. S. Christensen、R. Agarwal、E. De Carvalho 和 J. M. Cioffi，“使用加权最小均方误差进行加权和速率最大化的 MIMO-BC 波束成形设计”，IEEE《无线通信汇刊》，第7卷，第12期，第4792–4799页，2008年12月。[4] K. Shen 和 W. Yu，“通信系统的分式规划——第一部分：功率控制与波束成形”，IEEE《信号处理汇刊》，第66卷，第10期，第2616–2630页，2018年5月。[5] ——，“通信系统的分式规划——第二部分：通过匹配进行上行链路调度”，IEEE《信号处理汇刊》，第66卷，第10期，第2631–2644页，2018年5月。[6] Q. Hu、Y. Cai、Q. Shi、K. Xu、G. Yu 和 Z. Ding，“迭代算法诱导的深度展开神经网络：多用户 MIMO 系统的预编码设计”，IEEE《无线通信汇刊》，第20卷，第2期，第1394–1410页，2021年2月。[7] M. Zhu、T.-H. Chang 和 M. Hong，“异构大规模 MIMO 网络中的波束成形学习”，IEEE《无线通信汇刊》，第22卷，第7期，第4901–4915页，2023年7月。[8] L. Pellaco、M. Bengtsson 和 J. Jald´en，“加权最小均方误差波束成形算法的无矩阵求逆深度展开”，IEEE《通信学会开放期刊》，第3卷，第65–81页，2022年12月。[9] N. T. Nguyen、M. Ma、O. Lavi、N. Shlezinger、Y. C. Eldar、A. L. Swindlehurst 和 M. Juntti，“面向太赫兹大规模 MIMO 系统的深度展开混合波束成形设计”，IEEE《信号处理汇刊》，第71卷，第3788–3804页，2023年10月。[10] Q. Hu、Y. Liu、Y. Cai、G. Yu 和 Z. Ding，“联合深度强化学习与展开：基于透镜阵列的毫米波多用户 MIMO 波束选择与预编码”，IEEE《通信领域精选期刊》，第39卷，第8期，第2289–2304页，2021年8月。[11] C. Xu、Y. Jia、S. He、Y. Huang 和 D. Niyato，“基于深度展开技术的联合用户调度、基站聚类与波束成形设计”，IEEE《通信汇刊》，第71卷，第10期，第5831–5845页，2023年10月。[12] L. Schynol 和 M. Pesavento，“通过深度展开实现多小区多用户 MIMO 中的协同和速率最大化”，IEEE《通信领域精选期刊》，第41卷，第4期，第1120–1134页，2023年4月。[13] A. Chowdhury、G. Verma、A. Swami 和 S. Segarra，“多用户 MIMO 干扰网络中的深度图展开波束成形”，IEEE《无线通信汇刊》，第23卷，第5期，第4889–4903页，2024年5月。

[14] K. Shen、Z. Zhao、Y. Chen、Z. Zhang 和 H. V. Cheng，“加速二次变换和加权最小均方误差”，IEEE《通信领域精选期刊》，第42卷，第11期，第3110–3124页，2024年7月。[15] Z.-Q. Luo 和 S. Zhang，“动态频谱管理：复杂性与对偶性”，IEEE《信号处理精选主题期刊》，第2卷，第1期，第57–73页，2008年2月。[16] S. Joshi、P. C. Weeraddana、M. Codreanu 和 M. Latva-aho，“通过分支定界实现 MISO蜂窝下行网络的加权和速率最大化”，载于 2011年第45届阿西洛马会议记录：信号、系统与计算（阿西洛马会议），2011年4月，第1569–1573页。[17] L. Liu、R. Zhang 和 K.-C. Chua，“在 K用户多天线高斯干扰信道中实现加权和速率最大化的全局最优性”，IEEE《无线通信汇刊》，第11卷，第5期，第1933–1945页，2012年5月。[18] A. Kammoun、A. M¨uller、E. Bj¨ornson 和 M. Debbah，“基于多项式展开的线性预编码：大规模多小区MIMO系统”，IEEE《信号处理精选主题期刊》，第8卷，第5期，第861–875页，2014年10月。[19] X. Gao、O. Edfors、F. Rusek 和 F. Tufvesson，“实测超大规模MIMO信道中的线性预编码性能”，载于 2011年IEEE车辆技术会议（VTC秋季会议），2011年12月，第1–5页。[20] L. D. Nguyen、H. D. Tuan、T. Q. Duong 和 H. V. Poor，“多用户正则化迫零波束成形”，IEEE《信号处理汇刊》，第67卷，第11期，第2839–2853页，2019年6月。[21] D. P. Bertsekas，《非线性规划》。美国马萨诸塞州剑桥：麻省理工学院出版社，1999年。[22] X. Zhao、S. Lu、Q. Shi 和 Z.-Q. Luo，“重新思考WMMSE：其复杂度能否随 BS 天线数量线性扩展？”，IEEE《信号处理汇刊》，第71卷，第433–446页，2023年2月。[23] Z. Zhang、Z. Zhao 和 K. Shen，“通过下界最大化提升 WMMSE 和分式规划波束成形的效率”，载于 IEEE国际声学、语音与信号处理会议论文集（ICASSP），2023年5月。[24] R. Sun、C. Wang、A.-A. Lu、X. Fu、X. Liu、Y. Zhang、X. Gao 和 X.-G. Xia，“面向大规模MIMO下行链路的矩阵流形预编码器设计”，载于 IEEE无线通信与网络会议论文集（WCNC），2024年7月，第1–6页。[25] H. Sun、X. Chen、Q. Shi、M. Hong、X. Fu 和 N. D. Sidiropoulos，“学习优化：训练用于干扰管理的深度神经网络”，IEEE《信号处理汇刊》，第66卷，第20期，第5438–5453页，2018年10月。[26] W. Cui、K. Shen 和 W. Yu，“用于无线调度的空间深度学习”，IEEE《通信领域精选期刊》，第37卷，第6期，第1248–1261页，2019年6月。[27] W. Xia、G. Zheng、Y. Zhu、J. Zhang、J. Wang 和 A. P. Petropulu，“用于 MISO 下行链路波束成形优化的深度学习框架”，IEEE《通信汇刊》，第68卷，第3期，第1866–1880页，2020年3月。[28] H. He、C.-K. Wen、S. Jin 和 G. Y. Li，“基于深度学习的波束空间毫米波大规模MIMO系统信道估计”，IEEE《无线通信快报》，第7卷，第5期，第852–855页，2018年10月。[29] K. Gregor 和 Y. LeCun，“学习稀疏编码的快速近似方法”，载于国际机器学习会议（ICML），2010年6月，第399–406页。[30] J. R. Hershey、J. L. Roux 和 F. Weninger，“深度展开：新型深度架构的基于模型的启发”，2014年。 [在线]。可访问：https://arxiv.org/abs/ 1409 2574 。[31] A. Balatsoukas-Stimming 和 C. Studer，“通信系统的深度展开：综述与一些新方向”，载于 2019年IEEE信号处理系统研讨会（SiPS），2019年3月，第266–271页。[32] A. Goldsmith，《无线通信》。英国剑桥：剑桥大学出版社，2005年。[33] Y. Sun、P. Babu 和 D. P. Palomar，“信号处理、通信与机器学习中的主化-最小化算法”，IEEE《信号处理汇刊》，第65卷，第3期，第794–816页，2008年2月。[34] A. F. Agarap，“使用修正线性单元（ReLU）的深度学习”，2019［ ］ 1803 08375 。在线。可访问：[35] https://arxiv.org/abs/。H. Hojatian、J. Nadal、J.-F. Frigon 和 F. Leduc-Primeau，“用于大规模 MIMO 混合波束成形的无监督深度学习”，IEEE《无线通信汇刊》，第20卷，第11期，第7086–7099页，2021年5月。

Jianhang Zhu（IEEE学生会员）于2021年获得中国广州中山大学电子与信息技术学院通信工程工学学士学位，并于2024年获得中山大学计算机科学与工程学院计算机科学工学硕士学位。目前，他正在中国香港中文大学（深圳）理工学院攻读博士学位。他的研究兴趣包括语义通信、深度展开、非凸优化、机器学习、信息年龄、边缘计算和物联网。

Tsung-Hui Chang（IEEE会士）于2003年和2008年分别获得国立清华大学（NTHU）电气工程理学学士学位和通信工程博士学位。目前，他担任中国香港中文大学（深圳）（CUHK-SZ）人工智能学院教授兼副院长（教育），并担任广东省大数据计算重点实验室副主任。加入CUHK-SZ之前，他曾先后在国立台湾科技大学（NTUST）和加州大学戴维斯分校任职，分别担任教职人员和博士后研究员。他的研究兴趣包括数据通信和机器学习中的信号处理与优化问题。自2020年以来，他一直是IEEE信号处理学会（SPS）通信与网络信号处理技术委员会（SPCOM TC）的当选委员。他于2014年获得NTUST青年学者研究奖，2015年获得IEEE ComSoc亚太杰出青年研究员奖，2018年和2021年获得IEEE SPS最佳论文奖，2021年获得香港中文大学（深圳）SSE杰出教师研究奖，并于2024年获得香港中文大学（深圳）杰出研究奖。他是IEEE SPS一体化感知与通信技术工作组（ISAC TWG）的创始主席，并于2022年至2023年担任IEEE SPS理事会的当选大区理事。他曾在多家重要信号处理期刊的编委会任职，包括2014年至2018年担任IEEE信号处理汇刊副编委、2015年至2018年担任IEEE网络信号与信息处理汇刊副编委、2020年至今担任IEEE开放信号处理期刊副编委，以及2021年至2025年担任IEEE信号处理汇刊高级领域编辑。

Liyao Xiang（IEEE会员）于2012年获得中国上海上海交通大学电气与计算机工程工程学学士学位，并于2018年获得加拿大安大略省多伦多多伦多大学计算机工程博士学位。她目前是上海交通大学副教授。她的研究兴趣包括人工智能安全与隐私、数据挖掘中的隐私分析以及移动计算。

Kaiming Shen（IEEE高级会员）于2011年获得中国上海交通大学信息安全学士学位和数学理学学士学位，随后于2013年获得加拿大多伦多大学电气与计算机工程应用科学硕士学位。在渥太华的一家科技初创公司工作一年后，他返回多伦多大学，并于2020年初获得电气与计算机工程博士学位。自2020年起，他一直在中国香港中文大学（深圳）理工学院担任长聘轨助理教授。他的研究兴趣包括优化、无线通信和信息论。目前，他担任IEEE无线通信汇刊编委。他是IEEE信号处理学会通信与网络信号处理技术委员会成员。他于2021年获得IEEE信号处理学会青年作者最佳论文奖，2023年获得大学教学成果奖，2024年获得科学前沿奖，2025年获得中国信息论学会青年研究者奖。