学习快速、鲁棒地优化混合预编码

Ortal Lavi 和 Nir Shlezinger

arXiv:2301.00369v1 [eess.SP] 1 Jan 2023

摘要

混合预编码在以可控成本实现大规模多输入多输出（MIMO）发射机方面发挥着关键作用。MIMO预编码器需要根据信道条件的变化频繁调整。在混合MIMO中，预编码由数字与模拟波束成形组成，此类调整需要耗时的优化过程，并依赖准确的信道状态信息（CSI）。当信道快速变化以及使用含噪信道状态信息时，频谱效率会受到影响。在本研究中，我们采用深度学习技术，学习如何快速且鲁棒地优化混合预编码器，同时确保整个过程完全可解释。我们利用数据，为预设迭代次数的投影梯度和速率优化学习依赖于迭代次数的超参数设置。该算法将信道实现映射为混合预编码配置，同时保留优化器可解释的流程，并提升其收敛速度。为应对含噪信道状态信息，我们学习优化所有可容忍误差下的最小可达和速率，并基于投影概念镜像近端极小极大优化器提出一种鲁棒混合预编码方法。数值结果表明，与采用共享超参数的传统优化方法相比，我们的方法所需的迭代次数可减少十倍以上，同时实现相近甚至更优的和速率性能。

一、引言

无线通信网络在连接性、吞吐量和可靠性方面的需求不断增长。为满足这些需求，预计一项新兴技术将发挥关键作用：为无线基站（BS）配备大规模天线阵列，从而构建大规模多输入多输出（MIMO）网络[2]。尽管大规模多输入多输出的理论增益已得到充分确立 [3]–[5], 但以兼顾功耗与成本效益的方式实现此类大规模阵列，仍面临若干核心

本研究的部分成果曾以论文 [1] 的形式发表于2022年IEEE无线通信信号处理进展研讨会（SPAWC）。本研究部分由以色列创新局通过5G-WIN联盟资助。作者供职于内盖夫本-古里安大学电子与计算机工程学院（电子邮件：agivo@post.bgu.ac.il；nirshl@bgu.ac.il）。

挑战。其中之一是传统方案通常需要为每个天线单元配备专用射频链，而射频链成本高昂且功耗显著 [6]。

应对大规模多输入多输出在成本和功耗方面挑战的一种主流方法，是采用混合模拟/数字多输入多输出收发机 [7], [8]。混合多输入多输出收发机在模拟域中对部分发射和接收信号进行处理 [9]–[11]，因此所需的射频链数量少于天线数量。由此，混合MIMO发射机在数字域和模拟域中分别进行部分预编码。模拟处理受电路限制，通常使用矢量调制器 [12]或移相器 [9]实现。因此，与数字处理相比，模拟预编码通常受到更多限制；例如，数字处理通常可以在每个频率上应用不同的预编码器 [13]。

与成本高昂的全数字架构相比，混合多输入多输出的受限形式使得针对给定信道实现设定预编码模式变得更具挑战性。人们已提出多种方法来设计混合预编码系统，并优化其模拟处理和数字处理，以满足通信需求 [14]。常见方法是将预编码器的目标函数（例如和速率最大化，或最小化与全数字预编码器之间的距离 [15], [16], ）表述为优化问题。随后，使用迭代求解器处理所得优化问题；求解器会根据目标函数以及模拟电路和天线架构所带来的具体约束而有所不同。已有研究提出了用于调优混合多输入多输出系统的迭代算法，包括具有可控增益 [11], 、移相结构[10], [13], 、部分连接 [9], [15],、离散化矢量调制器 [12], 以及受洛伦兹约束的超表面天线 [17], [18]的系统。迭代优化器具有可解释性，因为它们是作为所构建问题的解推导出来的，但其收敛速度往往较慢。由于这种设定依赖瞬时信道状态信息（CSI），必须实时完成，以应对无线信道的频繁变化，而且其性能取决于CSI的准确性，因此收敛慢可能成为一项重大限制。

一种新兴的混合预编码器调优方法是基于深度学习的。该方法利用深度神经网络（DNN）学习复杂映射的能力，并可通过层数控制推理速度；与传统迭代优化器相比，其速度往往快得多 [19]。利用深度学习解决优化问题的方法称为学习以优化 [20]。 [21]–[27], 研究了深度神经网络辅助的混合预编码器优化，其中采用多层感知机 [21], [22]和卷积神经网络 [23]–[27] 作为优化器，而 [28], [29] 则采用深度

强化学习技术。虽然这类深度模型能够将信道估计映射为预编码器结构，但它们缺乏透明度或可解释性，无法说明输入数据如何转换为预编码器设置。此外，生成的预编码方案是针对用户数量以及训练所用信道的分布设计的。当信道分布或用户数量发生变化时，深度神经网络就需要重新训练，这一过程耗时较长。此外，参数量很大的深度神经网络可能计算复杂度过高，无法部署在硬件受限的无线通信设备上。

通过深度展开方法，可以从迭代优化算法中获得参数数量较少且可解释的深度模型 [30]。深度展开利用数据驱动的深度学习技术改进迭代优化器，而非用深度神经网络取代其运作 [31], [32]。深度展开优化器已用于配置 [33]中的单比特混合预编码器、 [34]中的全数字（非混合）窄带预编码器、 [35], [36]中的基于移相器的窄带混合预编码器，以及 [37]中具有有限反馈的窄带混合预编码器。后者使用生成网络展开优化算法以处理有限反馈，并使用过参数化残差网络优化预编码器，最终得到一个参数量很大的深度神经网络，这可能不适合应用于硬件受限的 MIMO 设备。文献 [38] 在具有完整 CSI 的混合 MIMO 发射机中，将专用深度神经网络集成到模拟预编码器的展开优化过程中，但推理时需要付出额外的复杂度。这促使我们设计可解释、轻量级的学习优化算法，以便快速将 CSI 转换为多频段混合预编码器，同时能够应对不准确的 CSI。

在本研究中，我们提出一种学习以优化算法，用于快速、鲁棒且可解释地调节多频段下行链路混合MIMO预编码器。我们考虑了多种模拟架构，包括具有可控增益的模拟预编码器，以及采用全连接移相器的架构。我们利用以过往信道实现形式提供的数据，加速传统迭代优化对可达和速率的收敛。为了设计一种展开算法，以应对 CSI 可能含噪这一具有挑战性且贴近实际的场景，我们首先研究可获得完整 CSI 的情况，并据此构建一种展开优化器，使其能够自然扩展到含噪 CSI 的场景。我们提出的数据辅助算法完整保留了其所源自的传统优化器的可解释性与灵活性，同时结构轻量，并且运行时间和复杂度固定。

具体而言，对于完整 CSI 场景下最大化可达和速率的问题，我们采用投影梯度上升（PGA）算法作为混合预编码设计优化器。我们采用这一方法的动机在于，它能够直接优化和速率，而不是采用某种更简单的替代目标函数，例如 [10], [15],，后者可由此扩展为对该指标进行鲁棒优化。我们通过固定较少的迭代次数来展开 PGA 算法，并且只训练与迭代相关的超参数。这些超参数可通过数据进行调节，同时不偏离算法的整体流程和运作方式。

我们选择以和速率目标函数下的 PGA 优化作为完整 CSI 混合预编码设计的基础，这有助于将我们的方法扩展到含噪 CSI 场景：我们发现，用于极大极小优化的投影概念镜像近端（PCMP）算法 [39] 适合作为含噪 CSI 下混合预编码的优化器。其算法步骤涉及计算和速率目标函数投影的梯度，与我们在完整 CSI 下使用 PGA 时所采用的方法相同。因此，我们提出通过将设计目标表述为：在给定的可容忍 CSI 误差范围内最大化最小可达和速率，来应对含噪 CSI。随后，我们对 PCMP 进行展开，得到一种可训练架构，其步骤与展开 PGA 时所用的步骤类似；同样，利用数据来调整优化器在每次迭代中的超参数。我们的实验结果评估了所提出的展开算法，并将其与源自这些算法的传统优化器进行比较，测试对象为多频段合成瑞利信道以及通过准确定性无线电信道生成器（QuaDRiGa）信道仿真器生成的信道 [40] ，涵盖多种信噪比（SNR）。我们系统地表明，所提出的展开算法在运行时所需的迭代次数和计算复杂度不到其迭代式对应算法的十分之一，同时能够达到相近甚至更高的和速率。

本文其余部分安排如下：第II节介绍系统模型；第III节推导了所提出的、利用完整 CSI 调节混合预编码器的学习以优化算法；第IV 节推导了适用于含噪信道状态信息的学习型优化器。我们在第V节对所提出的算法进行了数值评估，第VI 节给出了总结性意见。

本文中，向量用小写粗体字母表示，例如 $\boldsymbol{x}$； $[\boldsymbol{x}]_i$ 表示 $\boldsymbol{x}$的第i个元素。矩阵用大写粗体字母表示，例如 $\boldsymbol{X}$，其中$[\boldsymbol{X}]_{i,j}$ 是其第（i, j）个元素， $\boldsymbol{I}_n$表示 $n \times n$ 单位矩阵。集合用花体字母表示，例如 $\mathcal{X}$； $\mathbb{C}$ 是复数集合。转置、共轭转置、弗罗贝尼乌斯范数和随机期望分别表示为 $(\cdot)^T$、 $(\cdot)^H$、 $\|\cdot\|_F$和 $\mathrm{E}\{\cdot\}$。

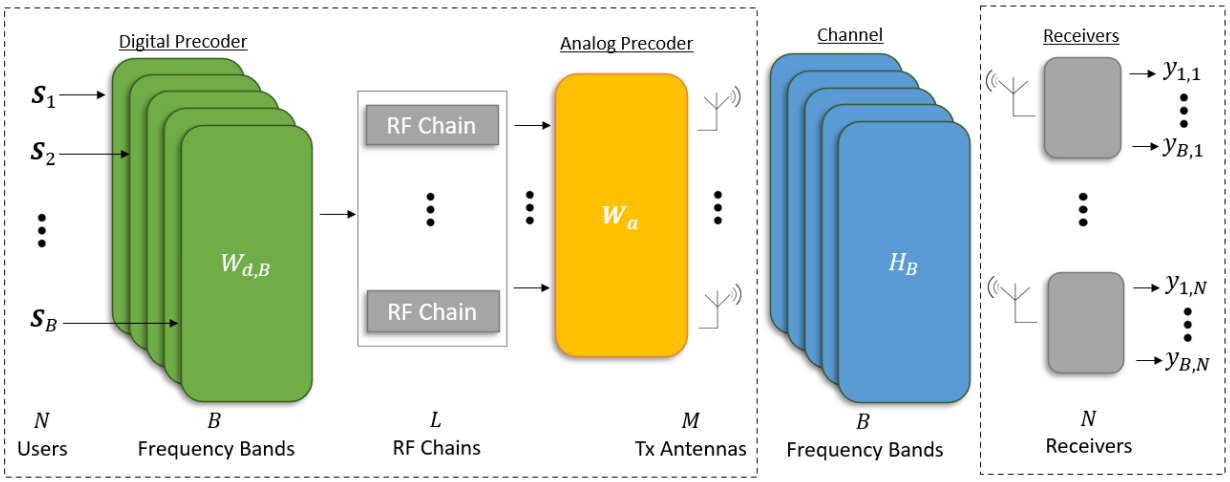


图 1：采用混合预编码的多输入多输出系统架构

II. 系统模型

本节介绍系统模型。我们首先在小节 II-A中回顾采用混合波束成形的宽带下行多输入多输出通信模型，随后在小节 II-B中讨论所考虑的模拟预编码器架构。然后，我们在小节 II-C 中阐述所考虑的快速稳健混合预编码器配置问题。

A. 采用混合波束成形的下行多输入多输出通信

1）信道模型： 我们考虑一个包含 N 名单天线用户的单小区下行混合MIMO系统。基站配备 M 根发射天线，并使用 B 个频带进行通信，所有用户以非正交方式共享频谱。整个系统如图1所示。

基站采用多载波信号传输，我们用 $\boldsymbol{s}_b \in \mathbb{C}^N$ 表示在第 b 个频率 bin（$b \in \{1, 2, \ldots, B\} \triangleq \mathcal{B}$）上传输的多用户发送信号向量。发送符号独立同分布且功率相等，因此对于每个 $b \in \mathcal{B}$，有 $\mathbb{E}[\boldsymbol{s}_b \boldsymbol{s}_b^H] = \frac{1}{N}\mathbf{I}_N$。对于每个频率 bin $b \in \mathcal{B}$，给定信道输入 $\boldsymbol{x}_b \in \mathbb{C}^M$，信道输出为

$$\boldsymbol{y}_b = \mathbf{H}_b \cdot \boldsymbol{x}_b + \boldsymbol{n}_b \in \mathbb{C}^N, \tag{1}$$

其中，$\mathbf{H}_b \in \mathbb{C}^{N \times M}$ 是第 b个频段子信道，而$\boldsymbol{n}_b \in \mathbb{C}^N$ 是元素独立同分布、方差为 σ^2的加性白高斯噪声（AWGN）。

2）混合波束成形：基站有 $L < M$ 条射频链，因此采用混合预编码。这里，多用户发送信号向量 $\{\boldsymbol{s}_b\}_{b \in \mathcal{B}}$ 分两个阶段进行预编码。首先，数字

预编码器 $\mathbf{W}_{d,b} \in \mathbb{C}^{L\times N}$ 作用于每个频率 bin $b \in \mathcal{B}$中的 $\boldsymbol{s}_b$，即，$\mathbf{W}_{d,b}$ 是第 b 个 bin 的数字预编码矩阵。接下来，数字符号经过 L 条射频链，并通过模拟预编码器合成为信道输入 $\boldsymbol{x}_b$。与数字处理不同，模拟预编码通过专用硬件实现，且假设其在频率上保持静态。因此，模拟预编码通过矩阵 $\mathbf{W}_a \in \mathcal{A} \subseteq \mathbb{C}^{M\times L}$建模，其中 $\mathcal{A}$ 表示可行的模拟预编码器设置集合，后文将对此进行讨论。

发射机在第 b个 bin 的输出为

$$\boldsymbol{x}_b = \mathbf{W}_a\mathbf{W}_{d,b}\cdot\boldsymbol{s}_b. \tag{2}$$

我们要求预编码器满足

$$\frac{1}{B}\sum_{j=1}^{B}\|\mathbf{W}_a\mathbf{W}_{d,j}\|_F^2 \le N, \tag{3}$$

作为发射机的总功率约束。将 (2) 代入 (1) 后，频率 bin N处的用户信道输出 b 可写为

$$\boldsymbol{y}_b = \mathbf{H}_b\mathbf{W}_a\mathbf{W}_{d,b}\cdot\boldsymbol{s}_b + \boldsymbol{n}_b \in \mathbb{C}^N. \tag{4}$$

B. 模拟预编码器

模拟预编码器$\mathbf{W}_a$可实现的映射（由集合$\mathcal{A}$表示）取决于硬件架构。我们考虑两种不同的模拟预编码器架构：无约束预编码器 Z 和全连接移相器网络。

1) 无约束预编码器： 此处，模拟预编码矩阵 $\mathbf{W}_a$不受硬件约束，可实现任意复矩阵（同时满足整体功率约束 (3)）。对于无约束预编码器，我们使用 $\mathcal{A} = \mathbb{C}^{M\times L}$。这种可配置衰减和移相的模拟处理可通过矢量调制器等方式实现，例如 [11]，[12]。虽然该表述除功率约束外不假设任何约束，但需要强调的是，无约束解对于受约束模拟硬件设计也很有用。这类设计通常会先求得无约束合并器，再将其投影到受约束集合中，例如 [10]。

2）移相器网络：模拟预编码器的一种常见实现方式是采用移相器 [9], [10]。在全连接移相器网络中，每个天线单元都连接

到每条射频链，并通过专用的可控移相器相连。这类预编码器可建模为元素幅度均为 1 的矩阵，即，

$$\mathcal{A}=\left\{\mathbf{A}\in\mathbb{C}^{M\times L}\middle|\left|[\mathbf{A}]_{m,l}\right|=1,\quad\forall(m,l)\right\}. \tag{5}$$

C. 问题构建

我们的目标是针对给定的信道实现设计混合预编码操作，以最大化多用户下行MIMO通信的可达和速率。将和速率定义为预编码目标函数是一种常见做法，[13], [18], [41], 因为和速率是衡量信道与混合预编码器综合作用效果的通信指标。我们重点研究两种场景——和速率最大化（给定准确的信道状态信息）和鲁棒和速率最大化（给定含噪信道状态信息）。

1）和速率最大化：由于 $[\boldsymbol{s}_b\boldsymbol{s}_b^H]=\frac{1}{N}\mathbf{I}_N$，可达到以下和速率 [42]

$$R\left(\mathbf{W}_a,\{\mathbf{W}_{d,b}\}_{b\in\mathcal{B}},\{\mathbf{H}_b\}_{b\in\mathcal{B}}\right)=\frac{1}{B}\sum_{b=1}^{B}\log\left|\mathbf{I}_N+\frac{1}{N\sigma^2}\mathbf{H}_b\mathbf{W}_a\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\mathbf{H}_b^H\right|. \tag{6}$$

因此，对于给定的信道实现 $\{\mathbf{H}_b\}_{b\in\mathcal{B}}$，我们的目标是求出 $\mathbf{W}_a^{(\mathrm{o})}$和 $\{\mathbf{W}_{d,b}^{(\mathrm{o})}\}_{b\in\mathcal{B}}$，使其成为以下优化问题的解

$$\left(\mathbf{W}_a^{(\mathrm{o})},\{\mathbf{W}_{d,b}^{(\mathrm{o})}\}_{b\in\mathcal{B}}\right)=\underset{\mathbf{W}_a,\{\mathbf{W}_{d,b}\}_{b\in\mathcal{B}}}{\operatorname{argmax}}\ R\left(\mathbf{W}_a,\{\mathbf{W}_{d,b}\},\{\mathbf{H}_b\}\right) \tag{7}$$

$$\text{s.t.}\quad\begin{cases}\frac{1}{B}\sum_{b=1}^{B}\|\mathbf{W}_a\mathbf{W}_{d,b}\|_F^2\le N\\ \mathbf{W}_a\in\mathcal{A}\end{cases}.$$

在（7）中，$\mathcal{A}$ 是上述模拟预编码模型对应的矩阵集合，即我们要在第II-B小节所述的不同约束条件下，求得合适的预编码器。

由于每当信道实现发生变化时都必须求解优化问题（7），因此我们提出的方案不仅应能求解（7），还应能快速完成，即在预先规定的计算量内完成。为实现这一目标，我们假设可以访问一组 $\mathcal{D}$ 先前遇到或模拟生成的信道实现 $\{\mathbf{H}^r\}_{r=1}^{|\mathcal{D}|}$，并利用它们在固定次数的运算内完成优化（运算次数由信道的相干时长等因素决定）。

2）鲁棒和速率最大化：优化问题（7）依赖于每个频率 bin 中信道的信息，即完整 CSI。若以（6）作为目标函数，当作为输入的信道矩阵是实际信道的（可能不准确的）估计值时，预计会影响混合预编码器的性能。因此，我们希望优化混合预编码器，使其能够适应信道状态信息中预先规定程度的误差。

为对此进行建模，我们考虑一个含噪子信道估计，记为 $\{\mathbf{H}_b + \mathbf{E}_b\}_{b\in\mathcal{B}}$，其中$\mathbf{H}_b$ 是真实的信道实现，而 $\mathbf{E}_b$ 是估计误差。我们希望设计混合预编码器，使其能够抵御预定义范围内的估计误差 ε，即当 $\|\mathbf{E}_b\|_F < \varepsilon$ 对每个 $b \in \mathcal{B}$都成立时。因此，我们将（7）转化为一个最大最小问题

$$\left(\mathbf{W}_a^{(\mathrm{o})}, \{\mathbf{W}_{d,b}^{(\mathrm{o})}\}_{b\in\mathcal{B}}\right) = \underset{\mathbf{W}_a, \{\mathbf{W}_{d,b}\}_{b\in\mathcal{B}}}{\operatorname{argmax}} \left\{ \min_{\|\mathbf{E}_b\|_F<\varepsilon} R\left(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b + \mathbf{E}_b\}\right) \right\} \tag{8}$$

$$\text{s.t.} \quad \begin{cases} \frac{1}{B}\sum_{b=1}^{B} \|\mathbf{W}_a \mathbf{W}_{d,b}\|_F^2 \le N \\ \mathbf{W}_a \in \mathcal{A} \end{cases} .$$

在（8）中，我们旨在最大化由可容忍的信道估计误差所导致的最小速率，即估计误差处于预定义范围 ε内。同样，我们希望快速执行鲁棒优化，并可利用以往的信道实现集合 $\mathcal{D}$ 来实现这一目标。

III. 基于全信道状态信息的快速混合预编码器学习优化

我们首先考虑在完整 CSI 条件下快速调整混合预编码器。对于无线信道 $\{\mathbf{H}_b\}_{b\in\mathcal{B}}$的每种实现，混合预编码器的配置都可以表述为一个约束优化问题（7）。因此，可以采用合适的迭代优化方法来设计混合预编码器。解决式（7）中优化问题的一种候选方法是 PGA。PGA 可以直接处理式（7）中的多载波和速率目标；而用于混合波束成形的其他迭代优化器则采用松弛目标，旨在逼近 [10], [15]中的全数字波束成形器。直接优化和速率而非其他替代目标函数（尽管优化该目标函数可能也有必要）的另一个动机是，由此得到的推导可以扩展到鲁棒优化，即处理含噪信道状态信息，详见第IV节。混合预编码器的配置取决于信道。因此，每当信道发生变化，也就是每个相干时长内，都必须执行一次优化过程；然而，PGA 等有理论依据的迭代优化器往往速度较慢、耗时较长。为应对这一挑战，本节介绍一种在完整

CSI 条件下学习优化混合预编码的方法，该方法专为快速配置混合预编码器而设计。我们的方法基于将 PGA 应用于式（7），具体内容见小节III-A；我们利用深度展开来优化其收敛速度，同时不牺牲可解释性和适用性 [31], ，推导过程见小节III-B ，并在小节III-C中进行讨论。

A. 投影梯度上升

问题（7）表示一个以 $B+1$ 优化矩阵变量$\mathbf{W}_a, \{\mathbf{W}_{d,b}\}_{b\in\mathcal{B}}$为变量的约束最大化问题。此类问题可以采用 PGA 算法结合交替优化来解决。在这种迭代方法中，每次迭代首先优化$\mathbf{W}_a$ ，同时保持$\{\mathbf{W}_{d,b}\}_{b\in\mathcal{B}}$ 固定，然后对每个$\mathbf{W}_{d,b}$重复这一过程。优化后的矩阵会经过投影，以确保不违反约束。

为了以数学形式表示这一操作，我们定义 $\tilde{\mathbf{H}}_b \triangleq \sqrt{\frac{1}{N\sigma^2}}\mathbf{H}_b$索引为$k+1$ 的每次迭代包含两个交替进行的阶段：模拟投影梯度上升和数字投影梯度上升。

模拟投影梯度上升：$\mathbf{W}_a$ 的更新按照以下方式进行

$$\mathbf{W}_a^{(k+1)} = \Pi_{\mathcal{A}}\left\{\mathbf{W}_a^{(k)} + \mu_a^{(k)}\frac{\partial}{\partial \mathbf{W}_a}R(\mathbf{W}_a^{(k)}, \{\mathbf{W}_{d,b}^{(k)}\}, \{\mathbf{H}_b\})\right\}, \tag{9}$$

其中， $\mu_a^{(k)}$是梯度步的步长，而 $\Pi_{\mathcal{A}}\{\cdot\}$ 是投影到集合 $\mathcal{A}$上的投影算子。关于R 的梯度相对于$\mathbf{W}_a$ 给出如下（见附录A）

$$\frac{\partial}{\partial \mathbf{W}_a}R(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\}) = \frac{1}{B}\sum_{b=1}^{B}\tilde{\mathbf{H}}_b^T\mathbf{G}_b(\mathbf{W}_a, \mathbf{W}_{d,b}, \mathbf{H}_b)^{-T}\tilde{\mathbf{H}}_b^*\mathbf{W}_a^*\mathbf{W}_{d,b}^*\mathbf{W}_{d,b}^T, \tag{10}$$

其中 $\mathbf{G}_b(\mathbf{W}_a, \mathbf{W}_{d,b}, \mathbf{H}_b) \triangleq (\mathbf{I}_N + \tilde{\mathbf{H}}_b\mathbf{W}_a\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_b^H)$。

投影算子 $\Pi_{\mathcal{A}}$ 在 (9) 中取决于可行模拟映射。对于无约束架构，显然 $\Pi_{\mathcal{A}}(\mathbf{A}) = \mathbf{A}$。对于基于全连接移相器的硬件，投影为

$$\Pi_{\mathcal{A}}\{\mathbf{A}\} = \tilde{\mathbf{A}}, \qquad [\tilde{\mathbf{A}}]_{m,l} = \frac{[\mathbf{A}]_{m,l}}{|[\mathbf{A}]_{m,l}|}, \forall(m,l). \tag{11}$$

数字投影梯度上升：模拟预编码器更新后，数字预编码器随之更新，且所有频率均并行执行更新。也就是说，每个 $b\in\mathcal{B}$ 的梯度步长计算如下

$$\hat{\mathbf{W}}_{d,b}^{(k+1)} = \mathbf{W}_{d,b}^{(k)} + \mu_{d,b}^{(k)}\frac{\partial}{\partial \mathbf{W}_{d,b}}R\Big(\mathbf{W}_a^{(k+1)}, \{\mathbf{W}_{d,b}^{(k)}\}, \{\mathbf{H}_b\}\Big), \tag{12}$$

其中 $\mu_{d\,b}^{(k)}$ ，为步长。对于每个R 相对于$\mathbf{W}_{d,\ b}$ 的梯度 $b \in \mathcal{B}$ 计算如下（参见附录A）

$$\frac{\partial}{\partial \mathbf{W}_{d,b}} R(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\}) = \frac{1}{B} \mathbf{W}_a^T \tilde{\mathbf{H}}_b^T \mathbf{G}_b (\mathbf{W}_a, \mathbf{W}_{d,b}, \mathbf{H}_b)^{-T} \tilde{\mathbf{H}}_b^* \mathbf{W}_a^* \mathbf{W}_{d,b}^*. \tag{13}$$

对所有频率区间执行梯度步骤后，通过以下方式将解投影到满足功率约束的范围内

$$\mathbf{W}_{d,b}^{(k+1)} = \sqrt{\frac{NB}{\sum_{b=1}^{B} \|\mathbf{W}_a^{(k+1)} \hat{\mathbf{W}}_{d,b}^{(k+1)}\|_F^2}} \cdot \hat{\mathbf{W}}_{d,b}^{(k+1)}. \tag{14}$$

整体流程总结为算法1。虽然 $\{\mathbf{W}_d^{(0)}$ 、$_b\}_{b\in\mathcal{B}}$的初始设置取为随机值，但初始模拟合并器$\mathbf{W}_a^{(0)}$被设为 L 的第一组右奇异向量，其中 $\frac{1}{B}\sum_{b=1}^{B} \tilde{}\mathbf{H}_b$ 表示频带子信道的平均值。这一设置对应于对（频率平均）信道的特征模进行模拟波束成形，是频率平坦多输入多输出信道的容量实现预编码方法的一部分 [43, Ch.10]。数值结果表明，这种有理论依据的初始化方式能够显著提升混合预编码器的性能；这些预编码器经过调优，可直接优化（7）中的速率。

算法1： 用于混合预编码的投影梯度上升

初始化： 将 $\{\mathbf{W}_d^{(0)}$ 随机化$\}_{b\in\mathcal{B}}$

$\mathbf{W}_a^{(0)} \leftarrow$ 的第一组 L 右奇异向量 $-$ $\tilde{\mathbf{H}}_b$

设置步长 $\{\mu_d^{(k)}{}_{,b}\}_{b\in\mathcal{B}}$, $\mu_a^{(k)}$

输入： 信道矩阵 $\{\tilde{\mathbf{H}}_b\}_{b\in\mathcal{B}}$

对于 $k = 0, 1, \ldots$ 直到收敛 **执行**

  更新 $\mathbf{W}_a^{(k+1)}$通过（9）

  对于 $b = 1, \ldots, B$ **do**

    计算 $\hat{}\mathbf{W}_d^{(k+1)}{}_{,b}$ 通过（12）

  end

  对于 $b = 1, \ldots, B$ **do**

    更新 $\mathbf{W}_d^{(k+1)}{}_{,b}$ 通过（14）

  end

结束

返回 $\{\mathbf{W}_d^{(k)}{}_{,b}\}_{b\in\mathcal{B}}$ 和 $\mathbf{W}_a^{(k)}$

基于梯度的优化器的收敛速度很大程度上取决于步长，即$\{(\{\mu_{d,b}^{(k)}\}_{b\in\mathcal{B}}, \mu_a^{(k)})\}$ 在算法1中。然而，基于线搜索和回溯等方法的传统步长优化方法 [44, 第 9]，通常需要额外的逐次迭代

计算，从而增加整体复杂度和运行时间。因此，常见做法是使用预先定义的手动调整常数步长，这可能导致收敛过程漫长。

B. 学习优化混合预编码

算法 1针对给定的信道实现优化混合预编码器。然而，其收敛速度在很大程度上取决于超参数，即步长，而步长往往难以设定。在此，我们提出利用深度学习中使用的自动化数据驱动优化器来调整依赖迭代的步长，即在较小且预先设定的迭代次数内学习优化混合预编码器。

我们的设计遵循深度展开方法 [30], [31], ，该方法将深度神经网络设计为具有固定迭代次数的迭代优化器。具体而言，我们采用算法 1 中的 PGA方法作为优化器，恰好进行 K 次迭代。这样，我们便能保证运行时间和复杂度精确可知且通常较小，从而满足实时要求。PGA 等一阶优化器的准确度通常不受超参数设置的影响，只要允许其运行至收敛（在步长足够小等温和条件下）；然而，当迭代次数固定时，其性能会受到这些参数的显著影响。因此，我们的设计将具有 K 次迭代的迭代优化器的超参数视为具有 K 层的深度神经网络的参数，并基于可用数据集 $\mathcal{D}$通过端到端训练对其进行调整，从而将 PGA 转换为可训练判别模型 [45]。

为此，将第 k次迭代的步长向量定义为 $\boldsymbol{\mu}_k \triangleq (\mu_a^{(k)}, \ldots, \mu_{d\,B}^{(k)})$，并将步长矩阵定义为 $\boldsymbol{\mu} \triangleq (\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{K-1})^T$，对应算法 K1 的次迭代。该 $K \times (B+1)$ 矩阵的元素是可训练参数，由数据学习得到。请注意，尽管 $\mathcal{D}$ 不包含真实预编码器，仍然可以进行端到端训练。这是因为可以使用式（6）中的可微度量评估混合预编码器的性能，并据此定义损失函数，以无监督方式训练超参数 $\boldsymbol{\mu}$ 。

$\tilde{}$ $\tilde{}$对于给定的归一化信道 $\mathbf{H} = \{\mathbf{H}_b\}_{b\in\mathcal{B}}$ 和步长 $\boldsymbol{\mu}$，损失函数计算为该信道（6）在$\tilde{}$每次 $\mathrm{PGA}_K(\mathbf{H}\ \boldsymbol{\mu})$ 迭代中所得可达和速率负值的加权平均。这意味着，当预编码器为 $\left(\mathbf{W}_a^{(k)}, \{\mathbf{W}_d^{(k)}{}_{,b}\}_{b\in\mathcal{B}}\right) \triangleq \mathrm{PGA}_k(\ \tilde{}\mathbf{H}\ \boldsymbol{\mu})$ 时，即经过 $k < K$ 次迭代并采用步长 $\boldsymbol{\mu}$，通过算法 1得到的预编码器对应的第 k次迭代和速率。由于每次迭代都需要提供一种设置，

使混合波束成形器在迭代过程中逐步得到改进，我们采用以下损失函数，其灵感来自 [46]

$$\mathcal{L}(\tilde{\mathbf{H}}, \boldsymbol{\mu}) = \frac{1}{K}\sum_{k=1}^{K}\log(1+k)\cdot\left(\frac{-1}{B}\sum_{b=1}^{B}\log\left|\mathbf{I}_N + \tilde{\mathbf{H}}_b\mathbf{W}_a^{(k)}\mathbf{W}_{d,b}^{(k)}\left(\tilde{\mathbf{H}}_b\mathbf{W}_a^{(k)}\mathbf{W}_{d,b}^{(k)}\right)^H\right|\right). \tag{15}$$

数据集 $\mathcal{D}$ 包含信道实现。由于 $\tilde{\mathbf{H}}_b \triangleq \sqrt{\frac{1}{N\sigma^2}}\mathbf{H}_b$ 具有已知的 N 和 σ^2，因此下文将数据集 $\mathcal{D}$ 中的条目写作 $\{\tilde{\mathbf{H}}^r\}_{r=1}^{|\mathcal{D}|}$。

学习优化方法旨在调整超参数向量 $\boldsymbol{\mu}$，使其最佳拟合数据集 $\mathcal{D}$，具体而言是使损失度量（15）最小。也就是说，我们的目标是将其设置为

$$\boldsymbol{\mu}^{(\mathrm{o})} = \arg\min_{\boldsymbol{\mu}} \frac{1}{|\mathcal{D}|}\sum_{r=1}^{|\mathcal{D}|}\mathcal{L}(\tilde{\mathbf{H}}^r, \boldsymbol{\mu}). \tag{16}$$

我们采用基于深度学习优化技术（例如16）的方法，利用小批量随机梯度下降等技术，根据数据集调整 $\boldsymbol{\mu}$ 基于数据集 $\mathcal{D}$。最终流程总结为算法2。在训练过程开始前，我们初始化 $\boldsymbol{\mu}$，并采用 PGA 收敛所需的固定步长。训练基于过去的信道实现，因此可以离线进行；训练完成后，学习得到的 $\boldsymbol{\mu}$ 将作为超参数，通过算法 K 1 的1快速将信道实现转换为混合预编码配置。最终的展开式PGA算法如图2所示。

算法2： 全信道状态信息下的学习优化混合预编码

初始化： 将 $\boldsymbol{\mu}$ 设为固定步长。

固定学习率 η

输入： 训练集 $\mathcal{D} = \{\tilde{\mathbf{H}}^r\}_{r=1}^{|\mathcal{D}|}$

个训练轮次 $= 0,\ 1,\ \ldots\ldots,$ 训练轮次 $_{\max} - 1$ **执行**

  将 $\mathcal{D}$ 随机划分为 Q个批次 $\{\mathcal{D}_q\}_{q=1}^{Q}$

  对于 $q = 1, \ldots, Q$ **执行**

    通过 $\mathbf{PGA}_K(\mathcal{D}_q, \boldsymbol{\mu})$ 计算预编码器

    计算该批次的平均损失： $\mathcal{L}(\boldsymbol{\mu}) = \frac{1}{|\mathcal{D}_q|}\sum_{\tilde{\mathbf{H}}\in\mathcal{D}_q}\mathcal{L}(\tilde{\mathbf{H}}\ \boldsymbol{\mu})$

    更新 $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \eta\nabla_{\boldsymbol{\mu}}\mathcal{L}(\boldsymbol{\mu})$

  end

结束

返回 $\boldsymbol{\mu}$

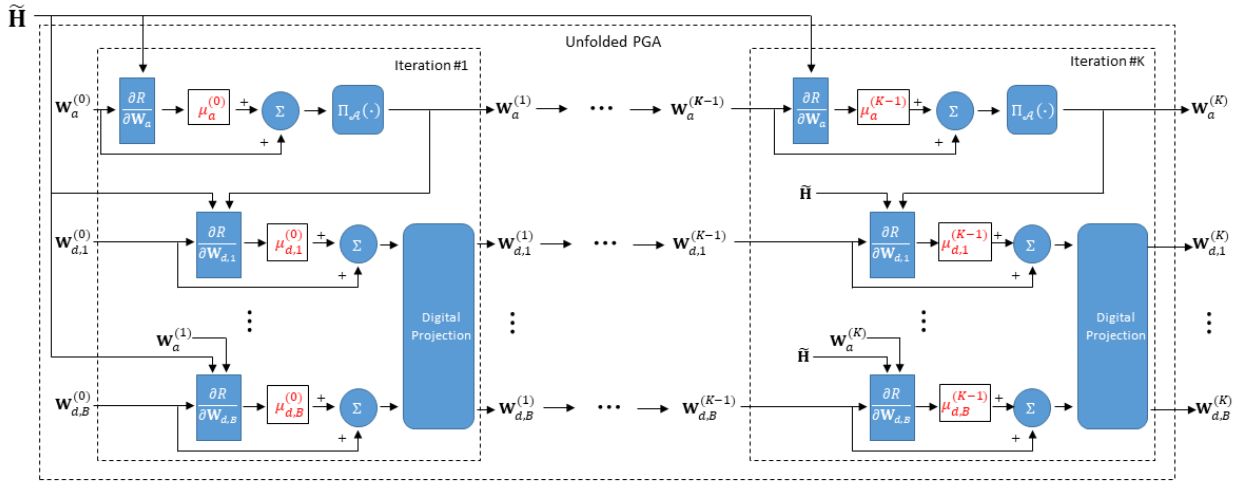


图2：展开式PGA模块示意图。学习得到的参数以红色字体标出。

C. 讨论

所提出的学习以优化方法利用数据提升基于迭代 PGA 的优化性能和收敛速度。由此得到的预编码器设计保留了经典 PGA 优化的可解释性和简洁性，同时能够像应用于此类任务的深度神经网络一样，以固定且较低的时延进行推断。因此，我们兼得基于模型的优化与数据驱动的深度学习之长。

迭代次数固定且有限这一事实，体现在为给定信道实现设置混合预编码器所涉及的计算复杂度上。为量化复杂度，我们考察算法1的一次迭代；其复杂度主要由第 $\mathcal{O}\big(BM^2(N+L)\big)$步的梯度计算（其复杂度阶为2 （与第 B 步的梯度计算复杂度阶相同12）以及第7步所需的投影（其复杂度阶为 $\mathcal{O}\big(BNML\big)$）主导。对于算法 K的1次迭代，且在 B 个频带上进行信号传输时，预定义 K 迭代次数的 PGA 算法（如所提出的学习型优化器所采用）的总体复杂度阶为

$$\mathcal{C}_{\mathrm{PGA}} = \mathcal{O}\Big(K \cdot BM\big(NL + M(N+L)\big)\Big). \tag{17}$$

我们将目标函数设为速率 $R(\cdot)$，因此，与计算替代目标函数（例如 [15] ）的梯度相比，计算其梯度每次迭代的复杂度更高（其二次依赖于射频链的数量，而非天线数量），但与 [16] 所需的复杂度处于相近的量级。不过，请记住，

我们对算法 2 的推导，是迈向快速且稳健优化速率的第一步；正如后文所示，这些梯度计算对此很有用。此外，所提出的学习优化框架便于实现迭代次数固定且较少的优化器，从而限制总体复杂度。

我们的方法遵循深度展开方法 [30]–[32]。不同于其他形式的深度展开网络（这类网络设计深度神经网络来模仿基于模型的优化器的运行方式，同时对其运行方式进行修改，例如 [46], ），我们的设计旨在保留基于模型的迭代优化的运行方式。我们利用深度学习工具的自动化训练能力来调整优化器的超参数。正如第V节所示，这样做改进了使用固定超参数的传统方式，也避免了在每次迭代中执行超参数搜索所带来的过长延迟。

我们在推导中采用的设计目标是可达和速率（6），该目标通过在渐近长度的码块上进行专用编码来逼近。尽管实际中可能会偏离（6），但它是整个信道的基本特征，涵盖了混合预编码、传输和接收，因此是优化混合MIMO系统的一项相关性能指标。此外，由于（6）中的目标函数明确且可微，我们的方法能够以无监督方式学习以优化，即无需像 [19],那样获取真实预编码器，也可以仅使用信道实现作为数据进行训练。然而，当作为输入的信道矩阵只是对真实信道的不准确估计时，使用（6）作为目标函数预计会影响混合预编码器的性能，其影响方式与此类信道状态信息误差对基于模型的优化器性能的影响相同。下一节将讨论如何扩展我们的方法以处理信道状态信息误差。

IV. 含噪信道状态信息下混合预编码器的快速稳健学习优化

在第III节中，我们假设已知完整的信道状态信息，并据此设计了混合预编码器。由此得到的算法2适用于特定信道，因此，当信道估计存在噪声时，预期会出现性能下降。对于失配信道状态信息，优化问题可表示为（8），其目标是最大化所有有界误差下的最小速率。本节将介绍一种方法，该方法基于我们对算法2 的推导，通过固定运行时间内的快速优化来求解（8）。我们的方法基于 PCMP 算法 [39], ，这是一种适用于（8）这类最大最小目标的迭代优化器。如第IV-A小节所述，PCMP 依赖投影梯度步，因此其推导可以利用通过 PGA 针对非鲁棒优化得到的步骤。接下来，按照

第III节所用的方法，我们采用深度展开，利用数据在固定迭代次数内优化其性能，如第IV-B小节所述。随后，我们将在第IV-C小节进行讨论。

A. 投影概念镜像近端法鲁棒优化

(8) 中的公式表示一个约束极大极小优化问题，其包含 $2B+1$优化（及辅助）矩阵变量 $\mathbf{W}_a$、 $\{\mathbf{W}_{d,\,b}\}_{b\in\mathcal{B}}$、 $\{\mathbf{E}_b\}_{b\in\mathcal{B}}$。这类问题可通过结合概念镜像近端（CMP）算法 [39] 与交替优化和投影来求解，从而得到 PCMP 方法。

PCMP 是一种迭代方法。每次迭代包含两个阶段：CMP 和投影。接下来，我们结合 (8) 来阐述这两个阶段。

CMP：根据模拟和数字预编码矩阵最大化 (8) 中的目标函数，并根据误差矩阵将其最小化。CMP 旨在通过梯度步迭代优化变量；在迭代索引 $k+1$ 下，梯度步的形式为

$$\begin{aligned}\left(\mathbf{W}_a^{(k+1)},\{\mathbf{W}_{d,b}^{(k+1)}\}_{b\in\mathcal{B}},\{\mathbf{E}_b^{(k+1)}\}_{b\in\mathcal{B}}\right)&=\left(\mathbf{W}_a^{(k)},\{\mathbf{W}_{d,b}^{(k)}\}_{b\in\mathcal{B}},\{\mathbf{E}_b^{(k)}\}_{b\in\mathcal{B}}\right)\\&+\mu^{(k)}\left(\frac{\partial}{\partial\mathbf{W}_a},\left\{\frac{\partial}{\partial\mathbf{W}_{d,b}}\right\},-\left\{\frac{\partial}{\partial\mathbf{E}_b}\right\}\right)R\left(\mathbf{W}_a^{(k+1)},\{\mathbf{W}_{d,b}^{(k+1)}\}_{b\in\mathcal{B}},\{\mathbf{H}_b+\mathbf{E}_b^{(k+1)}\}_{b\in\mathcal{B}}\right).\end{aligned}\tag{18}$$

(18) 中的计算通常无法直接实现，因为梯度是针对更新后的优化变量计算的（即迭代索引 $k+1$ 同时出现在更新方程的两侧）。不过，可以通过使用索引为 i 的额外迭代更新来实现，其形式为 [39, 公式 (6)]

$$\hat{\mathbf{W}}_{a,(i)}=\mathbf{W}_a^{(k)}+\mu_{a,(i)}^{(k)}\cdot\frac{\partial}{\partial\mathbf{W}_a}R\left(\hat{\mathbf{W}}_{a,(i-1)},\{\hat{\mathbf{W}}_{d,b,(i-1)}\},\{\mathbf{H}_b+\hat{\mathbf{E}}_{b,(i-1)}\}\right),\tag{19a}$$

$$\mathbf{W}_{d,b,(i)}=\mathbf{W}_{d,b}^{(k)}+\mu_{d,\,b\,(i)}^{(k)}\cdot\frac{\partial}{\partial\mathbf{W}_{d,b}}R\left(\hat{\mathbf{W}}_{a,(i-1)},\{\hat{\mathbf{W}}_{d,b,(i-1)}\},\{\mathbf{H}_b+\hat{\mathbf{E}}_{b,(i-1)}\}\right),\ \forall b\in\mathcal{B},\tag{19b}$$

$$\hat{\mathbf{E}}_{b,\,(i)}=\mathbf{E}_{\,b}^{(k)}-\mu_{e,\,b(i)}^{(k)}\cdot\frac{\partial}{\partial\mathbf{E}_b}R\left(\hat{\mathbf{W}}_{a,(i-1)},\{\hat{\mathbf{W}}_{d,b,(i-1)}\},\{\mathbf{H}_b+\hat{\mathbf{E}}_{b,(i-1)}\}\right),\forall b\in\mathcal{B}.\tag{19c}$$

在（19）中，$\mu_{a\,i\,()}^{(k)}$ $\mu_{d\,b\,,\,(i)}^{(k)}$ $\mu_{e\,b\,,\,(i)}^{(k)}$ 为步长。（19）中的优化变量初始化为

$$\left(\hat{\mathbf{W}}_{a,(0)},\{\hat{\mathbf{W}}_{d,b,(0)}\}_{b\in\mathcal{B}},\{\hat{\mathbf{E}}_{b,(0)}\}_{b\in\mathcal{B}}\right)=\left(\mathbf{W}_a^{(k)},\{\mathbf{W}_{d,b}^{(k)}\}_{b\in\mathcal{B}},\{\mathbf{E}_b^{(k)}\}_{b\in\mathcal{B}}\right).\tag{20}$$

（19）中关于模拟和数字预编码器的梯度计算，可以分别使用（10）和（13）中为 PGA 推导的梯度公式。对于每个R，关于$\mathbf{E}_b$ 的梯度 $b \in \mathcal{B}$ 可计算为（见附录B）

$$\frac{\partial}{\partial \mathbf{E}_b} R(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b + \mathbf{E}_b\}) = \frac{1}{B} \mathbf{G}_b(\mathbf{W}_a, \mathbf{W}_{d,b}, \mathbf{H}_b + \mathbf{E}_b)^{-T} (\tilde{\mathbf{H}}_b^* + \mathbf{E}_b^*) \mathbf{W}_a^* \mathbf{W}_{d,b}^* \mathbf{W}_{d,b}^T \mathbf{W}_a^T. \quad (21)$$

尽管上述 CMP 程序引入了一个额外的迭代过程，需要在索引为 k的每次迭代中执行，但通常只需对（19）执行两次迭代 [47],，即对（19）重复执行 $i = 1$，.....， $i_{\max}$ 其中 $i_{\max} = 2$。

投影：执行两次 CMP 后，将所得矩阵记为$\left(\hat{\mathbf{W}}_{a,\ (i_{\max})} \quad \{ \hat{\mathbf{W}}_{d,\ b,\ (i_{\max})} \}_{b \in \mathcal{B}}, \quad \{ \hat{\mathbf{E}}_{b,\ (i_{\max})} \}_{b \in \mathcal{B}} \right)$，并将其投影以满足约束。因此，迭代 $k+1$ 时的更新规则为

$$\mathbf{W}_a^{(k+1)} = \Pi_{\mathcal{A}} \left\{ \hat{\mathbf{W}}_{a,(i_{\max})} \right\}, \quad (22a)$$

$$\mathbf{W}_{d,b}^{(k+1)} = \sqrt{\frac{NB}{\sum_{b=1}^{B} \| \mathbf{W}_a^{(k+1)} \hat{\mathbf{W}}_{d,b,(i_{\max})} \|_F^2}} \cdot \hat{\mathbf{W}}_{d,b,(i_{\max})}, \quad \forall b \in \mathcal{B}, \quad (22b)$$

$$\mathbf{E}_b^{(k+1)} = \min \left\{ \frac{\varepsilon \cdot NM}{\| \hat{\mathbf{E}}_{b,(i_{\max})} \|_F}, 1 \right\} \cdot \hat{\mathbf{E}}_{b,(i_{\max})}, \quad \forall b \in \mathcal{B}, \quad (22c)$$

其中， ε 是子信道矩阵 $\{\mathbf{H}_b\}_{b \in \mathcal{B}} \in \mathbb{C}^{N \times M}$各元素的误差界。

整体过程总结于算法 3。矩阵 $\mathbf{W}_a^{(0)}, \{\mathbf{W}_d^{(0)}, _b\}_{b \in \mathcal{B}}$ 的初始化方式与算法1相同，而 $\{\mathbf{E}_b^{(0)}\}_{b \in \mathcal{B}}$ 则通过随机生成并归一化来保证 $\|\mathbf{E}_b\|_F < \epsilon$ 对每个 $b \in \mathcal{B}$均成立。请注意，算法 3 描述的是一种基于梯度的优化器，与算法1类似。因此，其收敛速度同样取决于步长 $\mu_a^{(k)}$ () $\{\mu_{d,b}^{(k)}(i)\}_{b \in \mathcal{B}}, \{\mu_{e,b}^{(k)}(i)\}_{b \in \mathcal{B}}$，而这些步长难以手动选取。在 IV-B 小节中，我们将介绍快速学习优化方法，并将算法2中针对完整 CSI 使用的思路加以扩展，通过数据辅助的超参数调优来加速算法 3 的收敛。

B. 鲁棒学习优化混合预编码

算法 3 针对给定的含噪信道实现优化混合预编码器。然而，每当信道发生变化时，都需要实时执行该算法，因此仍需在固定且较少的迭代次数内快速获得可靠的混合预编码器。为此，我们采用深度展开方法，遵循以下方案

A算法 3： 鲁棒混合预编码的投影概念镜像近端法

初始化：随机生成 $\{\mathrm{W}_d^{(0)},{}_b\}_{b\in\mathcal{B}}, \{\mathrm{E}_b^{(0)}\}_{b\in\mathcal{B}}\mathrm{W}_a^{(0)} \leftarrow$ 中的前 L 个右奇异向量 $\frac{1}{B}\sum_b \tilde{}\mathrm{H}_b$，设置步长 $\neg\mu_{a\ i}^{(k)}(\ )$ $\{\mu_{d,b}^{(k)}$ $(\tilde{}\ {}_i\}_{b\in\mathcal{B}}$ $\{\mu_{e,b\ i}^{(k)}\}_{b\in\mathcal{B}})$，$(\ )$ 输入：信道矩阵 $\{\mathrm{H}_b\}_{b\in\mathcal{B}}$1 对于 $k=0$，1，.....，直到收敛为止 2 设置 (20) 3 对于 $i=1$，.....，$i_{\max}$ 执行 4 根据 (19) 计算 $\left(\hat{}\mathrm{W}_{a,(i)}, \{\hat{}\mathrm{W}_{d,b,(i)}\}_{b\in\mathcal{B}}, \{\hat{}\mathrm{E}_{b,(i)}\}_{b\in\mathcal{B}}\right)$ 5 结束 6 根据 (22) 更新 $\left(\mathrm{W}_a^{(k+1)}, \{\mathrm{W}_d^{(k+1)},{}_b\ \}_{b\in\mathcal{B}}, \{\mathrm{E}_b^{(k+1)}\}_{b\in\mathcal{B}}\right)$ 7 结束 8 返回 $\{\mathrm{W}_d^{(k)},{}_b\}_{b\in\mathcal{B}}$ 和 $\mathrm{W}_a^{(k)}$

如第 III-B小节所述，利用其在涉及速率函数投影梯度的优化中调整超参数的能力，二者的主要区别在于学习参数的数量和算法结构。

我们使用 PCMP方法作为优化器，迭代次数恰好为 K，内部迭代次数为 $i_{\max}=2$。定义第 k 次迭代中模拟更新的步长向量为 $\boldsymbol{\mu}_a^k \triangleq (\mu_{a\ 1}^{(k)}(\ )\ \mu_{a\ 2}^{(k)}(\ ))^T$；第 k 次迭代中数字更新的步长向量为 $\{\boldsymbol{\mu}_{d,b}^k\}_{b\in\mathcal{B}} \triangleq \{(\mu_{d\ b}^{(k)}, (1)\ \mu_{d\ b}^{(k)}, (2))^T\}_{b\in\mathcal{B}}$；误差更新的步长向量为 $\{\boldsymbol{\mu}_{e,b}^k\}_{b\in\mathcal{B}} \triangleq \{(\mu_{e\ b}^{(k)}, (1)\ \mu_{e\ b}^{(k)}, (2))^T\}_{b\in\mathcal{B}}$。相应地，将第 k 次迭代的步长矩阵定义为 $\boldsymbol{\mu}^k \triangleq (\boldsymbol{\mu}_a^k, \boldsymbol{\mu}_{d,1}^k, \ldots, \boldsymbol{\mu}_{d,B}^k, \boldsymbol{\mu}_{e,1}^k, \ldots, \boldsymbol{\mu}_{e,B}^k)$，从而得到算法 3 进行 K 次迭代时的步长张量 $\boldsymbol{\mu} \triangleq (\boldsymbol{\mu}^0, \ldots, \boldsymbol{\mu}^{K-1})$。

展开架构将经过 K 次迭代的 PCMP方法转换为一种判别式算法，其可训练参数为 $\boldsymbol{\mu}$ 的各项。与 III-B 小节采用的方法类似，我们以无监督方式训练展开式 PCMP，即数据集仅包含信道实现，而可容忍误差水平 ε 为给定值。对于给定的归一化信道 $\mathrm{H} = \{\mathrm{H}_b\}_{b\in\mathcal{B}}$ 和步长 $\boldsymbol{\mu}$，其损失函数 $\tilde{}\quad\ \tilde{}$ 定义为该信道所得可达和速率的最大负值 (6)，其中最大值是在所有$\|\mathrm{E}_b\|_F < \varepsilon$上取的，预编码器为 $\left(\mathrm{W}_a^{(K)}、\{\mathrm{W}_d^{(K)}、{}_b\}_{b\in\mathcal{B}}\right) \triangleq \mathrm{PCMP}_K(\ \tilde{}\mathrm{H}\ \boldsymbol{\mu})$，即通过算法 3 经 K 次迭代并采用步长 $\boldsymbol{\mu}$ 得到的预编码器。由此得到的损失为

$$\mathcal{L}(\tilde{\mathbf{H}}, \boldsymbol{\mu}) = \max_{\|\mathbf{E}_b\|_F<\varepsilon} \left\{ \frac{-1}{B}\sum_{b=1}^{B} \log\left| \mathbf{I}_N + \frac{1}{N\sigma^2}(\mathbf{H}_b+\mathbf{E}_b)\mathbf{W}_a^{(K)}\mathbf{W}_{d,b}^{(K)}\left((\mathbf{H}_b+\mathbf{E}_b)\mathbf{W}_a^{(K)}\mathbf{W}_{d,b}^{(K)}\right)^H \right| \right\}. \quad (23)$$

式（23）中的最大化项显著增加了将其用作学习训练目标的难度

PCMP 超参数。为克服这一问题，我们通过随机生成得到一个由 n_e 误差模式（满足 $\|\mathbf{E}_b\|_F < \varepsilon$）组成的有限集合，并将零误差（即 $\mathcal{E} \triangleq \left\{\{\mathbf{E}_b^t\}_{b\in\mathcal{B}}\right\}_{t=1}^{n_e} \cup \{\mathbf{0}\}$）加入其中，然后寻找使这些误差模式中的最大损失最小化的设置。也就是说，为确保训练可行，用于训练展开算法的目标函数根据其经过 K 次迭代后的输出计算损失，表示为

$$\tilde{\mathcal{L}}(\tilde{\mathbf{H}}, \boldsymbol{\mu}) = \max_{\mathbf{E}_b \in \mathcal{E}} \left\{ \frac{-1}{B} \sum_{b=1}^{B} \log \left| \mathbf{I}_N + \frac{1}{N\sigma^2} (\mathbf{H}_b + \mathbf{E}_b) \mathbf{W}_a^{(K)} \mathbf{W}_{d,b}^{(K)} \left((\mathbf{H}_b + \mathbf{E}_b) \mathbf{W}_a^{(K)} \mathbf{W}_{d,b}^{(K)} \right)^H \right| \right\}. \quad (24)$$

鲁棒学习优化方法（总结于算法 4）使用数据来调整 $\boldsymbol{\mu}$。我们在训练过程开始前初始化 $\boldsymbol{\mu}$，并使用固定步长，使 PCMP 收敛（尽管不一定能够快速收敛）。训练完成后，将学习得到的矩阵 $\boldsymbol{\mu}$ 作为超参数，通过执行算法 K 3 的迭代，将含噪信道实现快速转换为混合预编码配置。

算法 4： 鲁棒学习优化混合预编码

初始化： 将 $\boldsymbol{\mu}$ 设为固定步长。固定学习率 η

输入： 训练集 $\mathcal{D} = \{\tilde{\mathbf{H}}^r\}_{r=1}^{|\mathcal{D}|}$

对于 训练轮次 $= 0, 1, \ldots\ldots,$ 训练轮次 $_{\max} - 1$ **执行**

  将 $\mathcal{D}$ 随机划分为 Q 个批次 $\{\mathcal{D}_q\}_{q=1}^{Q}$

  对于 $q = 1, \ldots, Q$ **执行**

    通过 $\mathbf{PCMP}_K(\mathcal{D}_q, \boldsymbol{\mu})$ 计算预编码器

    计算该批次的平均损失：$\mathcal{L}(\boldsymbol{\mu}) = \frac{1}{|\mathcal{D}_q|} \sum_{\tilde{\mathbf{H}} \in \mathcal{D}_q} \tilde{\mathcal{L}}(\tilde{\mathbf{H}}, \boldsymbol{\mu})$

    更新 $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \eta \nabla_{\boldsymbol{\mu}} \mathcal{L}(\boldsymbol{\mu})$

  end

结束

返回 $\boldsymbol{\mu}$

C. 讨论

算法4 中提出的鲁棒学习优化方法2 中的方法，以应对含噪信道状态信息。具体而言，对于特殊情况 $\varepsilon = 0$ 和 $i_{\max} = 1$，PCMP算法（算法3）会退化为PGA方法（算法1）。相应地，利用数据优化后的算法4 是算法2的特例。因此，我们采用渐进式设计，首先在第III节中设计了应用于速率函数的学习式PGA优化，并在此基础上拓展推导，构建了用于最大最小速率优化的鲁棒学习式PCMP算法。此外，这一实现表明，通用的

学习优化方法可用于加速广泛的迭代式、依赖超参数的基于模型算法的收敛，尤其适用于采用梯度和投影步骤的算法。

就复杂度而言，具有 K 次迭代和 $i_{\max}=2$ 的展开式 PCMP，其复杂度阶与采用完整 CSI 的 PGA 方法相同。具体而言，算法3 的第4步在每次迭代中重复计算算法 1 $i_{\max}$ 次梯度，并额外计算关于 $\mathbf{E}_b$ 的梯度（其复杂度与计算关于$\mathbf{W}_{d、b}$的梯度相同）。因此，展开式 PCMP 算法的复杂度阶为

$$\mathcal{C}_{\text{PCMP}}=\mathcal{O}\Big(K\cdot BM\big(NL+M(N+L)i_{\max}\big)\Big), \tag{25}$$

对于 $i_{\max}=2$，其复杂度与式（17）中的展开式 PGA 相近。

通过展开式 PCMP 实现快速且稳健的优化，可得到对信道估计误差不太敏感的混合预编码器。PCMP算法通过在设置混合预编码器时考虑范围广泛的信道误差，从而应对失配信道。PGA 算法会在信道每次变化时运行，而 PCMP算法更加稳健、稳定，因此运行频率可以更低。因此，PCMP 也可用于降低优化过程的执行频率，而不仅仅是应对失配信道。

误差界值 ε是鲁棒算法中的关键因素。它会影响解决方案的鲁棒性、可靠性和稳定性，因此需要依据一些先验知识或应用需求来选择。另一种方法是在进行本文所述的预编码器设计之前，采用信道估计方法，以此克服信道不匹配问题。另一个重要参数是在训练期间评估的预期误差模式的设定，即 $\mathcal{E}$。在我们的评估中，我们以随机方式生成这一集合，并将零误差模式纳入其中，以确保其适用于无误差情况；此外，也可以考虑其他方法来构建该集合，使得到的优化后的超参数最适合预期的鲁棒优化度量。由于此设定在线下完成，因此复杂度并非关键问题，可以考虑采用序贯采样和贝叶斯优化技术 [48] 来设定 $\mathcal{E}$。我们将把对本方法扩展的研究留待未来开展。

V. 数值评估

接下来，我们将对所提出的展开式优化框架进行数值评估[1]。我们的主要目的是通过数值结果证明，所提出的学习优化框架能够显著减少迭代次数，相较于采用固定超参数的传统优化方法尤为如此。

我们的数值评估首先从第 V-A 小节中完整 CSI 的设置开始。由于完整 CSI 设置（模拟合并器受限于实现移相器网络）是混合预编码中常见的设置，因此本次数值评估可让我们将设计的第一步，即展开式PGA方法，与现有优化方法进行比较。接着，我们在第 V-B 小节中，考虑基于展开式PCMP的鲁棒优化。对于这两种 CSI 设置，我们按照第 II-A 小节所述，模拟下行混合MIMO系统的两种不同模型：瑞利衰落，即每个频率 bin 中的信道矩阵均从独立同分布高斯分布中随机生成；以及 QuaDRiGa 模型 [40], ，这是一种开源的基于几何的随机信道模型，适用于模拟真实的 MIMO 系统。

A. 全信道状态信息下的混合预编码

对于完整 CSI，用于设计混合预编码器的优化器是 PGA，其数据辅助学习优化版本详见第III-B小节。该方法既可用于无约束模拟合并器（其唯一约束是功率约束），也可用于硬件受限设计，其中常见的约束是移相器网络。因此，我们将评估分为两部分：采用无约束模拟合并器的学习式 PGA，以及采用移相器的 PGA。

1) 使用无约束模拟合并器的 *PGA*：在这种情况下，我们使用模拟投影算子 $\Pi_{\mathcal{A}}(\mathbf{A}) = \mathbf{A}$，即不对模拟架构施加任何约束。这类无约束合并器可以使用例如 [11], [12] 中提出的架构进行近似。

我们考虑三种不同的设置，分别涉及频带数量 B、用户数量N、射频链数量 L、发射天线数量 M以及数据源：1) 配置为 $B = 8, N = 6, L = 10, M = 12$ 的瑞利信道，包含 8 个频率区间；2) 包含 $B = 16$ 个频率区间的 QuaDRiGa信道，其中使用 $N = 4, L = 6, M = 12$；3) 包含 $B = 128, N = 4, L = 8, M = 16$ 的宽带 QuaDRiGa信道。

对于每种配置，我们应用算法 2来学习如何仅用

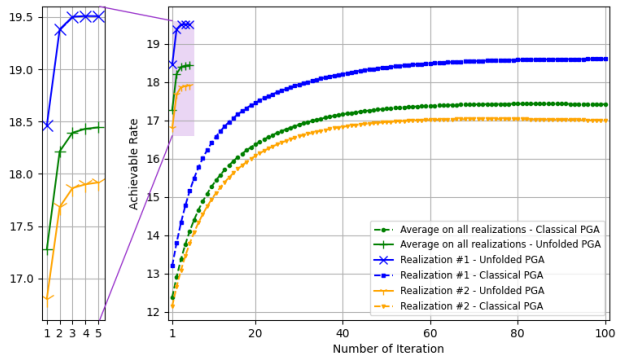
[1]完整源代码可在 https://github.com/ortalagiv/Learn-to-Rapidly-Optimize-Hybrid-Precoding。

$K = 5$，10 次迭代，基于 $|\mathcal{D}| = 1000$ 信道实现，其中我们使用了 $50 - 70$ 个训练轮次，批量大小为 $|\mathcal{D}_q| = 100$，并使用 Adam 更新算法6 中算法2的步骤。

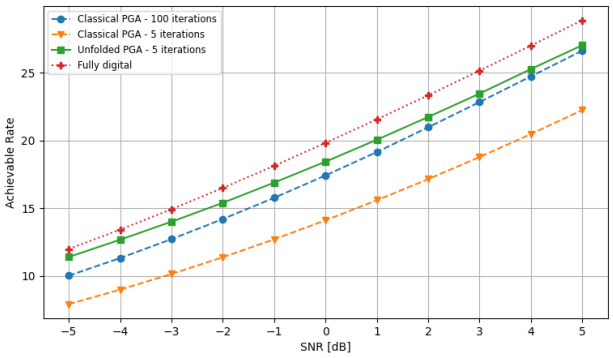
将经过 K 优化迭代的 PGA 混合预编码设计与采用固定超参数的 PGA 进行比较，后者使用根据经验试验选定的恒定步长。我们在 100 个未见过的测试信道上评估这两种混合预编码器。图 3、图 4 和图 5 分别展示了设置 $(B = 8, N = 6, L = 10, M = 12)$、$(B = 16, N = 4, L = 6, M = 12)$ 和 $(B = 128, N = 4, L = 8, M = 16)$ 下的仿真结果。每幅图既比较了算法在所有信道实现上的平均收敛情况及随机选取的信道实现上的收敛情况，也展示了所得和速率随信噪比的变化。

所得到的和速率（信噪比为$=$ 0dB 时）随 PGA 迭代次数的变化情况如图 3a、图 4a 和图 5a 所示。图中给出了展开式 PGA 和经典 PGA（采用人工选定的恒定步长）在所有未见信道上的平均结果，以及两个随机信道实现上的结果。我们观察到，所提出的学习以优化方法始终能够促进混合预编码器的简单 PGA 优化，以远少于传统 PGA 所需的迭代次数，达到相近甚至更高的和速率。与采用固定步长的传统优化相比，学习以优化使迭代次数减少了 20（5 次迭代，而非 100 次迭代）和 10（10 次迭代，而非 100次迭代），分别实现相应的加速。这些提升在所考虑的不同信道模型和配置下均保持一致。图 3b、图 4b 和图 5b 展示了所提出的展开式 PGA 在不同信噪比下的和速率，并与经典 PGA 的和速率以及全数字基带预编码所得的和速率进行比较；后者作为可达和速率的上界。这些图表明，将经典 PGA 算法转化为判别式可训练模型能够带来增益。具体而言，展开算法仅通过少数几次迭代实现 PGA，同时持续优于采用固定步长并迭代 100 次的传统 PGA。这些结果表明，所提出的方法能够利用数据提升性能和收敛速度，同时保留传统迭代优化器的可解释性和适用性。

2）采用移相器网络的 *PGA*：上述数值评估表明，所考虑的深度展开方法确实有助于快速优化混合预编码器。为证明这些优势并非无约束模拟合并器所独有，并与其他迭代优化器进行比较，接下来我们考虑用于

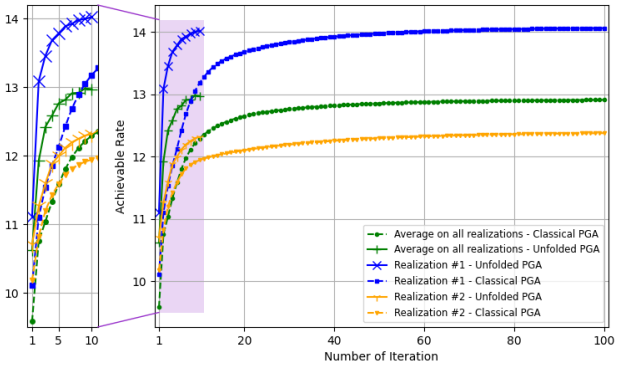


(a) 每次 PGA 迭代的和速率。

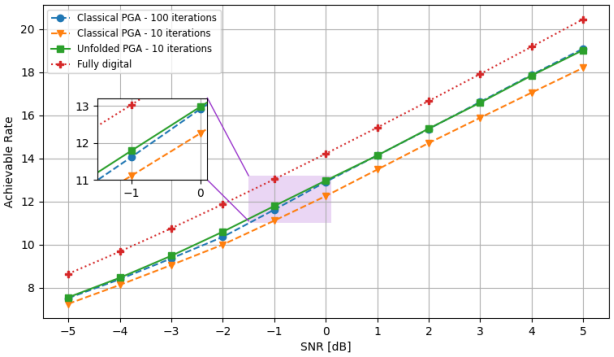


(b) 和速率与信噪比的关系。

图 3：瑞利 信道；无约束模拟预编码器； $B = 8,\ N = 6,\ L = 10,\ M = 12$.

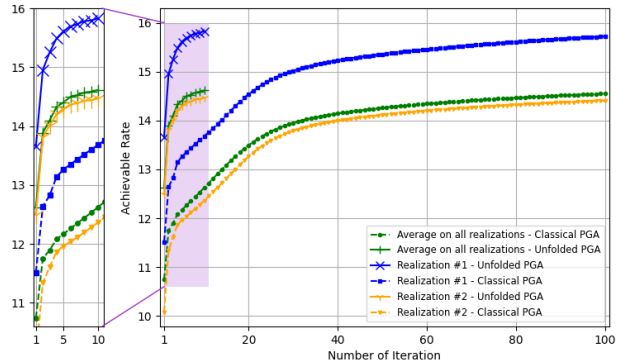


(a) 每次 PGA 迭代的和速率。

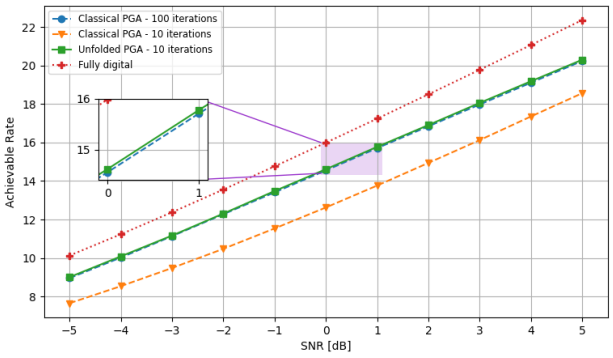


(b) 和速率与信噪比的关系。

图 4：QuaDR iGa信道；无约束模拟预编码器； $B = 16,\ N = 4,\ L = 6,\ M = 1$ 2.



(a) 每次 PGA 迭代的和速率。



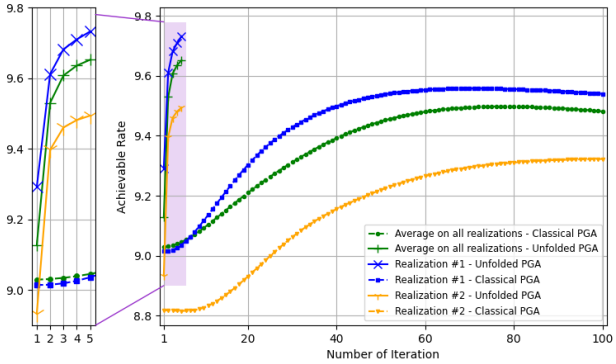
(b) 和速率与信噪比的关系。

图 5：QuaDRiGa信道；无约束模拟预编码器； $B = 128,\ N = 4,\ L = 8,\ M = 16$。

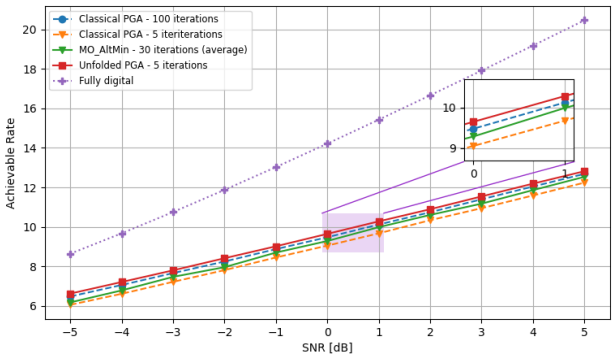
模拟架构的实现方式，这也是文献中常见的做法。这意味着在算法1的步骤2 中使用式（11）中的投影。我们使用 [15], 中的 MO-AltMin 算法作为基准，该算法旨在迭代优化此类受约束混合预编码器。

我们考虑 QuaDRiGa 信道模型，并针对频带数量 B、用户数量 N、射频链数量 L 和发射天线数量M采用两种不同设置：1) $B = 16, N = 4, L = 2, M = 12$；以及 2) $B = 128, N = 4, L = 2, M = 16$。与无约束情形类似，对于每种设置，我们应用算法2来学习超参数，使 PGA 在经过 $K = 5$次迭代后收敛。在训练过程中，每种设置使用 $|\mathcal{D}| = 1000$ 个信道，并训练 50 轮，批量大小为 $|\mathcal{D}_q| = 100$ ；步骤 6 中的更新采用 Adam。我们将经过 $K = 5$ 次迭代学习得到的 PGA 与采用经验选定固定超参数的 PGA 进行比较。我们在 100 个未见过的测试信道上评估混合预编码器优化器。性能评估结果如图6和图7所示，分别对应设置 B =16, $N = 4, L = 2, M = 12$ 和 $B = 128$, $N = 4, L = 2, M = 16$；每幅图都考察了收敛曲线，以及优化过程结束时达到的和速率随信噪比的变化。

信噪比= 0dB 时的和速率，取所有未见信道以及两个随机信道实现的平均值，随 PGA 迭代次数的变化情况如图6a和图 7a所示。从这些图中可以看出，与采用恒定且人工选定步长的标准 PGA 相比，学习优化方法能够加速 PGA 优化并取得更优表现。值得注意的是，迭代次数减少了 20 (5 次迭代，而标准优化需要100次），相较标准优化大幅减少。数据驱动优化能够帮助应对由此产生的优化问题的非凸特性，因此系统性地实现了性能提升。图6b和图 7b 展示了所提出的展开式 PGA、经典PGA、文献 [15], 中的 MO-AltMin算法以及全数字基带预编码之间的比较。结果表明，由于混合架构中射频链数量相对较少，全数字基带预编码与混合预编码之间存在显著差距。比较这些迭代算法可见，它们的和速率差异较小；但需要注意的是，所提出的展开式 PGA 所需迭代次数最少，因此相较于基准方法和标准 PGA，速度有所提升。这些结果表明，所提出的学习优化技术能够处理并将不同约束纳入 PGA 算法，同时实现快速性能提升并完全保留可解释性。

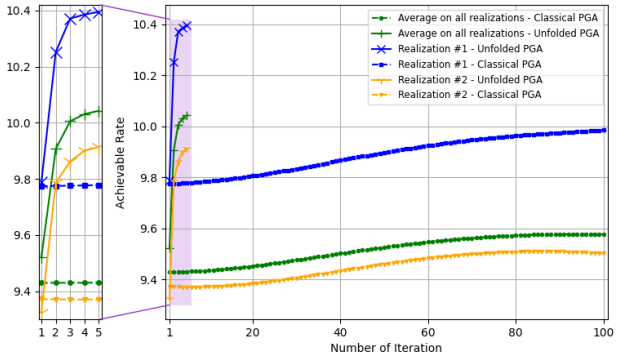


(a) 每次 PGA 迭代的和速率。

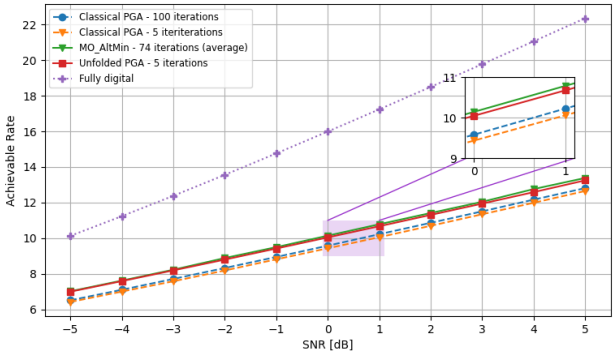


(b) 和速率与信噪比的关系。

图6：QuaDRiGa信道；受约束模拟预编码器； $B = 16$， $N = 4$， $L = 2$， $M = 12$。



(a) 每次 PGA 迭代的和速率。



(b) 和速率与信噪比的关系。

图7：QuaD RiGa 信道；受约束模拟预编码器； $B = 128$， $N = 4$， $L = 2$， $M = 16$.

B. 含噪信道状态信息下的混合预编码

V-A 小节报告的数值研究通过实证验证了所提出的方法能够在不同设置下实现快速优化。接下来，我们将展示该方法同时实现快速且鲁棒的混合预编码的能力。因此，以下将考虑展开 PCMP算法，以便在存在噪声信道估计的情况下进行预编码。在展示了我们的方法处理受约束预编码器的能力后，此处我们将重点关注无约束模拟合并器。相应地，我们将学习优化方法（即算法 4）用于算法 3；其中，步骤6 中应用于模拟预编码器的投影算子设为 $\Pi_{\mathcal{A}}(\mathbf{A})$ =**A**。所考虑的误差模式集合 $\mathcal{E}$ 由以下内容构成： $n_e = 20$ 个随机生成的模式，其弗罗贝尼乌斯范数值范围为$(0, \epsilon)$，并在 $\mathcal{E}$上评估极大极小速率。

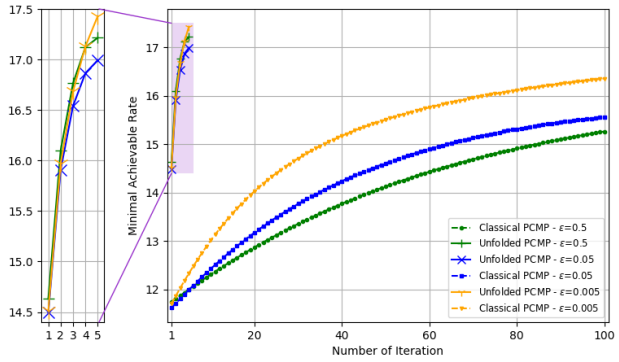
我们考察两种 在频带数量 B、用户数量 N、射频链

L、发射天线数量 M 以及数据源方面： $B = 8, N = 6, L = 10, M = 12$，采用瑞利信道；而 $B = 16, N = 4, L = 6, M = 12$ 采用 QuaDRiGa 信道模型。每种设置下的仿真结果分别展示在图8和图9中。同样，我们评估了优化过程结束时，随信噪比变化的收敛速率和最小速率（处于可容忍误差范围内），并将结果与传统的 PCMP优化器进行比较。仿真中考虑了三个误差界值： $\varepsilon = 0.005$、0.05、0.5。对于每种设置，我们应用算法 4，基于 $|\mathcal{D}| = 1000$ 个信道，通过 $K = 5$ 次迭代学习如何设置混合预编码器；训练时使用了 $40 - 50$ 个训练轮次，批量大小为 $|\mathcal{D}_q| = 100$，并在步骤 6 中使用 Adam 进行更新。我们将采用恒定超参数的标准 PCMP 与经过优化且恰好迭代 $K = 5$ 次的 PCMP 进行了比较。

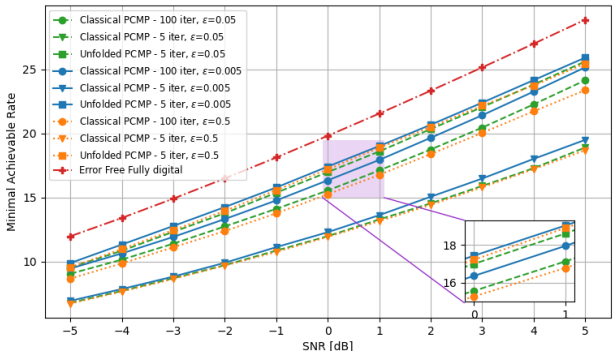
图 8a 和图 9a 展示了三种误差界值下，针对 100 个未见测试信道，求得的最小和速率平均值随 PCMP 迭代次数的变化。在这些图中，我们观察到收敛速度显著提升，达到 20 倍（5 次迭代对比 100 次迭代）；结果还表明，学习得到的 PCMP 算法性能始终优于采用恒定步长的传统 PCMP。图 8b和图 9b展示了不同信噪比下的和速率。这些图还包含无误差的全数字基带预编码和速率，该速率根据完整 CSI 计算，反映了在已知完整 CSI 且不进行射频链路缩减的情况下所能达到的最佳性能。我们观察到，所提出的鲁棒优化方法——展开式 PCMP——取得的性能与基于完整 CSI 的全数字基带预编码相当接近，并且始终优于需要更多迭代次数的基于模型的 PCMP。这些结果展示了我们提出的学习优化方法在实现快速且鲁棒的混合预编码器优化方面的优势。

VI. 结论

在本研究中，我们提出了一种利用数据实现混合预编码器快速、稳健且可解释调优的方法。我们的方法将一个适用于在给定可容忍信道状态信息误差范围内最大化最小和速率的优化器，展开为固定且数量较少的迭代次数。随后，我们利用数据调优每次迭代的超参数。研究表明，该方法能够显著提高收敛速度，同时配置出与通过耗时的非学习型优化调优得到的混合预编码器相比，实现相近甚至更高和速率的混合预编码器。

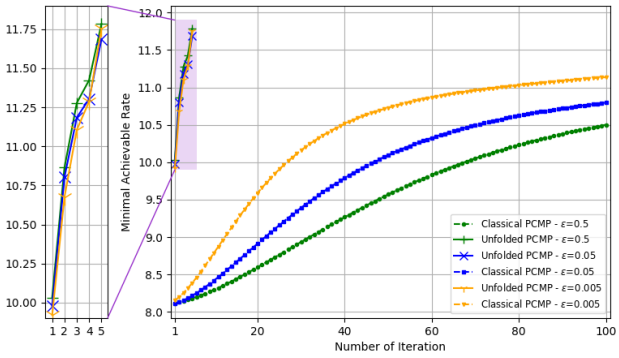


(a) 每次 PCMP 迭代的最小和速率。

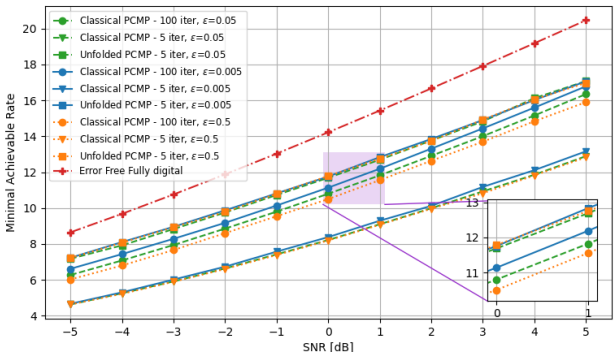


(b) 最小和速率与信噪比的关系。

图8：瑞利信道；无约束模拟预编码器； $B = 8,\ N = 6,\ L = 10,\ M = 12$。



(a) 每次 PCMP 迭代的最小和速率。



(b) 最小和速率与信噪比的关系。

图9：QuaD RiGa 信道；无约束模拟预编码器； $B = 16,\ N = 4,\ L = 6,\ M = 1$ 2.

VII. 致谢

作者谨此感谢 CEVA 的 Tomer Yeblonka 提供的宝贵意见和富有意义的讨论。

附录 A

关于 $R(\cdot)$ 相对于 $\mathbf{W}_a$， $\{\mathbf{W}_{d,\ b}\}_{b\in\mathcal{B}}$ 的复梯度

我们首先推导（10），方法是按照 [49] 中详细介绍的通用标量函数复值梯度计算方法进行计算。关于 $\mathbf{R}\,(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\})$ 关于 $\mathbf{W}_a$ 的复微分

is

$$
\begin{aligned}
&d_{\mathbf{W}_a} R\left(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\}\right) \\
&= d_{\mathbf{W}_a}\left(\frac{1}{B}\sum_{b=1}^{B}\log\left|\mathbf{I}_N + \tilde{\mathbf{H}}_b\mathbf{W}_a\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_b^H\right|\right) \\
&= \mathrm{Tr}\Big[\frac{1}{B}\sum_{b=1}^{B}\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_b^H\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b)^{-1}\tilde{\mathbf{H}}_b(d\mathbf{W}_a)\Big] \\
&\quad + \mathrm{Tr}\Big[\frac{1}{B}\sum_{b=1}^{B}\mathbf{W}_{d,b}^*\mathbf{W}_{d,b}^T\mathbf{W}_a^T\tilde{\mathbf{H}}_b^T\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b)^{-1}\tilde{\mathbf{H}}_b^*(d\mathbf{W}_a^*)\Big].
\end{aligned}
$$

使用 [49, 表 3。2], 由此可得（10）。

同样，$R\left(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\}\right)$关于$\mathbf{W}_{d,b}$的微分为

$$
\begin{aligned}
&d_{\mathbf{W}_{d,b}} R\left(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b\}\right) \\
&= d_{\mathbf{W}_{d,b}}\left(\frac{1}{B}\sum_{n=1}^{B}\log\left|\mathbf{I}_N + \tilde{\mathbf{H}}_n\mathbf{W}_a\mathbf{W}_{d,n}\mathbf{W}_{d,n}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_n^H\right|\right) \\
&= \mathrm{Tr}\Big[\frac{1}{B}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_b^H\tilde{\mathbf{H}}_b\mathbf{W}_a(d\mathbf{W}_{d,b})\Big] \\
&\quad + \mathrm{Tr}\Big[\frac{1}{B}\mathbf{W}_{d,b}^T\mathbf{W}_a^T\tilde{\mathbf{H}}_b^T\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b)^{-T}\tilde{\mathbf{H}}_b^*\mathbf{W}_a^*(d\mathbf{W}_{d,b}^*)\Big].
\end{aligned}
$$

利用 [49, 表 3。2], 可得（13）。

APPENDIX B
COMPLEX GRADIENT OF $R(\cdot)$ WITH RESPECT TO $\{\mathbf{E}_b\}_{b\in\mathcal{B}}$

$R(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b + \mathbf{E}_b\})$关于$\mathbf{E}_b$的复微分为

$$
\begin{aligned}
&d_{\mathbf{E}_b} R(\mathbf{W}_a, \{\mathbf{W}_{d,b}\}, \{\mathbf{H}_b + \mathbf{E}_b\}) \\
&= d_{\mathbf{E}_b}\left(\frac{1}{B}\sum_{n=1}^{B}\log\left|\mathbf{I}_N + \left(\tilde{\mathbf{H}}_n + \mathbf{E}_n\right)\mathbf{W}_a\mathbf{W}_{d,n}\mathbf{W}_{d,n}^H\mathbf{W}_a^H\left(\tilde{\mathbf{H}}_n + \mathbf{E}_n\right)^H\right|\right) \\
&= \mathrm{Tr}\Big[\frac{1}{B}\Big(\mathbf{W}_a\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\tilde{\mathbf{H}}_b^H\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b+\mathbf{E}_b)^{-1} \\
&\quad + \mathbf{W}_a\mathbf{W}_{d,b}\mathbf{W}_{d,b}^H\mathbf{W}_a^H\mathbf{E}_b^H\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b+\mathbf{E}_b)^{-1}\Big)(d\mathbf{E}_b)\Big] \\
&\quad + \mathrm{Tr}\Big[\frac{1}{B}\Big(\mathbf{W}_a^*\mathbf{W}_{d,b}^*\mathbf{W}_{d,b}^T\mathbf{W}_a^T\tilde{\mathbf{H}}_b^T\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b+\mathbf{E}_b)^{-T} \\
&\quad + \mathbf{W}_a^*\mathbf{W}_{d,b}^*\mathbf{W}_{d,b}^T\mathbf{W}_a^T\mathbf{E}_b^T\mathbf{G}_b(\mathbf{W}_a,\mathbf{W}_{d,b},\mathbf{H}_b+\mathbf{E}_b)^{-T}\Big)(d\mathbf{E}_b^*)\Big].
\end{aligned}
$$

使用 [49, 表 3。2], 由此得到（21）。

参考文献

[1] O. Agiv 和 N. Shlezinger，“学习快速优化混合预编码”，载于 IEEE SPAWC 会议论文集，2022 年。[2] “6G——面向所有人的下一代超互联体验”，三星，2020 年。[3] T. L. Marzetta，“具有无限数量基站天线的非协作蜂窝无线通信”，IEEE 无线通信汇刊，第 9 卷，第 11 期，第 3590 页，2010 年。[4] E. Bj¨ornson、J. Hoydis 和 L. Sanguinetti，“大规模 MIMO 网络：频谱、能源与硬件效率”，《信号处理基础与趋势》，第 11 卷，第 3–4 期，第 154–655 页，2017 年。[5] N. Shlezinger 和 Y. C. Eldar，“非协作上行链路大规模 MIMO 系统的频谱效率”，IEEE 通信汇刊，第 67 卷，第 3 期，第 1956–1971 页，2019 年。[6] X. Gao、L. Dai 和 A. M. Sayeed，“支持 5G 无线通信的大天线阵列毫米波 MIMO 低射频复杂度技术”，IEEE 通信杂志，第 56 卷，第 4 期，第 211–217 页，2018 年。[7] A. F. Molisch、V. V. Ratnam、S. Han、Z. Li、S. L. H. Nguyen、L. Li 和 K. Haneda，“大规模 MIMO 的混合波束成形：综述”，IEEE 通信杂志，第 55 卷，第 9 期，第 134–141 页，2017 年。[8] I. Ahmed、H. Khammari、A. Shahid、A. Musa、K. S. Kim、E. De Poorter 和 I. Moerman，“5G 混合波束成形技术综述：架构与系统模型视角”，IEEE 通信调查与教程，第 20 卷，第 4 期，第 3060–3097 页，2018 年。[9] R. M´endez-Rial、C. Rusu、N. Gonz´alez-Prelcic、A. Alkhateeb 和 R. W. Heath，“毫米波通信的混合 MIMO 架构：移相器还是开关？”，IEEE Access，第 4 卷，第 247–267 页，2016 年。[10] S. S. Ioushua 和 Y. C. Eldar，“适用于大规模 MIMO 系统的一类混合模拟–数字波束成形方法”，IEEE信号处理汇刊，第 67 卷，第 12 期，第 3243–3257 页，2019 年。[11] T. Gong、N. Shlezinger、S. S. Ioushua、M. Namer、Z. Yang 和 Y. C. Eldar，“MIMO 系统的射频链路缩减：硬件原型”，IEEE 系统期刊，第 14 卷，第 4 期，第 5296–5307 页，2020 年。[12] T. Zirtiloglu、N. Shlezinger、Y. C. Eldar 和 R. T. Yazicigil，“采用低分辨率模数转换器和任务专用波束成形的高能效混合 MIMO 接收机”，载于 IEEE ICASSP 会议论文集，2022 年，第 5338–5342 页。[13] S. Park、A. Alkhateeb 和 R. W. Heath，“宽带毫米波 MIMO 系统混合预编码的动态子阵列”，IEEE 无线通信汇刊，第 16 卷，第 5 期，第 2907–2920 页，2017 年。[14] M. A. Albreem、A. H. Al Habbash、A. M. Abu-Hudrouss 和 S. S. Ikki，“大规模 MIMO 预编码技术概述”，IEEE Access，第 9 卷，第 60 764–60 801 页，2021 年。[15] X. Yu、J.-C. Shen、J. Zhang 和 K. B. Letaief，“毫米波 MIMO 系统混合预编码的交替最小化算法”，IEEE 信号处理专题期刊，第 10 卷，第 3 期，第 485–500 页，2016 年。[16] F. Sohrabi 和 W. Yu，“大规模天线阵列的混合数字与模拟波束成形设计”，IEEE 信号处理专题期刊，第 10 卷，第 3 期，2016 年。[17] N. Shlezinger、G. C. Alexandropoulos、M. F. Imani、Y. C. Eldar 和 D. R. Smith，“面向 6G 极大规模 MIMO 通信的动态超表面天线”，IEEE 通信杂志，第 28 卷，第 2 期，第 106–113 页，2021 年。[18] H. Zhang、N. Shlezinger、F. Guidi、D. Dardari、M. F. Imani 和 Y. C. Eldar，“近场多用户 MIMO 通信的波束聚焦”，IEEE 无线通信汇刊，第 21 卷，第 9 期，第 7476–7490 页，2022 年。[19] A. Zappone、M. Di Renzo 和 M. Debbah，“深度学习时代的无线网络设计：基于模型、基于人工智能，还是两者兼有？”，IEEE 通信汇刊，第 67 卷，第 10 期，第 7331–7376 页，2019 年。[20] T. Chen、X. Chen、W. Chen、H. Heaton、J. Liu、Z. Wang 和 W. Yin，“学习优化：入门指南与基准测试”，arXiv 预印本 arXiv:2103.12828，2021 年。[21] Q. Hu、Y. Cai、K. Kang、G. Yu、J. Hoydis 和 Y. C. Eldar，“用于信道获取与混合预编码的双时间尺度端到端学习”，IEEE 信号处理专题期刊，第 40 卷，第 1 期，第 163–181 页，2021 年。

[22] H. Huang、Y. Song、J. Yang、G. Gui 和 F. Adachi，“基于深度学习的毫米波大规模多输入多输出混合预编码”，IEEE《车辆技术汇刊》，第 68 卷，第 3 期，第 3027–3032 页，2019 年。[23] H. Huang、Y. Peng、J. Yang、W. Xia 和 G. Gui，“基于深度学习的快速波束成形设计”，IEEE《车辆技术汇刊》，第 69 卷，第 1 期，第 1065–1069 页，2019 年。[24] A. M. Elbir 和 A. K. Papazafeiropoulos，“多用户毫米波大规模MIMO系统的混合预编码：一种深度学习方法”，IEEE《车辆技术汇刊》，第 69 卷，第 1 期，第 552–563 页，2019 年。[25] A. M. Elbir、K. V. Mishra 和 S. Chatzinotas，“太赫兹频段联合超大规模MIMO雷达通信：基于模型与无模型的混合波束成形”，IEEE《信号处理专题选刊》，第 15 卷，第 6 期，第 1468–1483 页，2021 年。[26] T. Peken、S. Adiga、R. Tandon 和 T. Bose，“用于奇异值分解和混合波束成形的深度学习”，IEEE《无线通信汇刊》，第 19 卷，第 10 期，第 6621–6642 页，2020 年。[27] 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。[28] Q. Hu、Y. Liu、Y. Cai、G. Yu 和 Z. Ding，“深度强化学习与展开联合：采用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。[29] Q. Wang、K. Feng、X. Li 和 S. Jin，“PrecoderNet：基于深度强化学习的毫米波系统混合波束成形”，IEEE《通信快报》，第 9 卷，第 10 期，第 1677–1681 页，2020 年。[30] N. Shlezinger、J. Whang、Y. C. Eldar 和 A. G. Dimakis，“基于模型的深度学习”，arXiv 预印本 arXiv:2012.08405，2020 年。[31] N. Shlezinger、Y. C. Eldar 和 S. P. Boyd，“基于模型的深度学习：深度学习与优化的交汇”，IEEE Access，第 10 卷，第 115 384–115398 页，2022 年。[32] A. Balatsoukas-Stimming 和 C. Studer，“面向通信系统的深度展开：综述与一些新方向”，载于 IEEE SiPS 会议论文集，2019 年，第 266–271 页。[33] A. Balatsoukas-Stimming、O. Casta˜neda、S. Jacobsson、G. Durisi 和 C. Studer，“神经网络优化的1比特预编码：面向大规模多用户MIMO”，载于 IEEE SPAWC 会议论文集，2019 年。[34] L. Pellaco、M. Bengtsson 和 J. J ald´en，“加权MMSE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，2021 年。[35] J. Luo、J. Fan 和 J. Zhang，“MDL-AltMin：结合深度学习与交替优化的毫米波系统混合预编码方案”，IEEE《通信快报》，第 11 卷，第 9 期，第 1925–1929 页，2022 年。[36] S. Shi、Y. Cai、Q. Hu、B. Champagne 和 L. Hanzo，“基于符号错误概率最小化的深度展开神经网络辅助混合波束成形”，IEEE《车辆技术汇刊》，2022 年，提前在线发表。[37] E. Balevi 和 J. G. Andrews，“采用生成对抗网络压缩超低反馈开销的展开式混合波束成形”，IEEE《无线通信汇刊》，第 20 卷，第 12 期，第 8381–8392 页，2021 年。[38] K.-Y. Chen、H.-Y. Chang、R. Y. Chang 和 W.-H. Chung，“ 通过深度展开实现毫米波 MIMO-OFDM 系统中的混合波束成形 ”，载于车辆技术会议：（ VTC2022-Spring）。IEEE，2022 年。[39] K. K. Thekumparampil、P. Jain、P. Netrapalli 和 S. Oh，“光滑极小极大优化的高效算法”，《神经信息处理系统进展》，第 32 卷，2019 年。[40] S. Jaeckel、L. Raschkowski、K. B¨ orner 和 L. Thiele，“QuaDRiGa：一种具有时间演进功能、用于实现虚拟现场试验的三维多小区信道模型”，IEEE《天线与传播汇刊》，第 62 卷，第 6 期，第 3242–3256 页，2014 年。[41] X. Gao、L. Dai、S. Han、I. Chih-Lin 和 R. W. Heath，“面向大规模天线阵列毫米波MIMO系统的能效混合模拟数字预编码”，IEEE《通信领域精选期刊》，第 34 卷，第 4 期，第 998–1009 页，2016 年。[42] Z. Shen、R. Chen、J. G. Andrews、R. W. Heath 和 B. L. Evans，“采用块对角化的多用户MIMO广播信道和容量”，IEEE《无线通信汇刊》，第 6 卷，第 6 期，第 2040–2045 页，2007 年。

[43] A. Goldsmith，《无线通信》。剑桥，2005 年。[44] S. Boyd 和 L. Vandenberghe，《凸优化》。剑桥，2004 年。[45] N. Shlezinger 和 T. Routtenberg，“随机信号线性估计的判别式与生成式学习 [讲义],”*arXiv* 预印本 *arXiv:2206.04432*，2022 年。[46] N. Samuel、T. Diskin 和 A. Wiesel，“学习检测”，*IEEE*信号处理汇刊，第 67 卷，第 10 期，第 2554–2564 页，2019 年。[47] A. Mokhtari、A. Ozdaglar 和 S. Pattathil，“鞍点问题中超梯度法与乐观梯度法的统一分析：近端点方法”，载于国际人工智能与统计会议。PMLR，2020 年，第 1497–1507 页。[48] P. I. Frazier，“贝叶斯优化教程”，*arXiv* 预印本 *arXiv:1807.02811*，2018 年。[49] A. Hj¨orungnes，《复值矩阵导数及其在信号处理和通信中的应用》。剑桥，2011 年。