

基于深度展开的大规模 MIMO 系统 CSI 反馈算法

廖勇, 程港, 李玉杰

(重庆大学微电子与通信工程学院, 重庆 400044)

摘要: 针对现阶段大规模 MIMO 系统中基于深度学习的信道状态信息 (CSI) 反馈算法待训练参数过多、可解释性不强的问题, 提出了 2 种基于深度展开的 CSI 反馈算法。一种是基于可学习参数的近似消息传递 (AMP) 算法, 该算法利用深度学习中的可学习参数将 AMP 算法中阈值函数的阈值和 Onsager 校正项的参数替换, 增强了阈值函数在应对非严格稀疏数据时的非线性能力。另一种是基于卷积网络的 AMP 算法, 该算法将阈值函数模块替换为卷积残差学习模块, 利用该模块去除 AMP 算法中每轮迭代产生的高斯随机噪声。仿真分析表明, 所提算法具有比 AMP 算法更好的 CSI 反馈表现, 其中基于卷积网络的 AMP 算法具有比基于深度学习的代表性方法更优异的 CSI 重构性能。

关键词: CSI 反馈; 深度学习; 深度展开; 近似消息传递; 可学习参数; 卷积网络

中图分类号: TN929.5

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2022237

CSI feedback algorithm based on deep unfolding for massive MIMO systems

LIAO Yong, CHENG Gang, LI Yujie

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

Abstract: In order to solve the problem that the channel state information (CSI) feedback algorithm based on deep learning in massive MIMO systems at present had too many parameters to be trained and could not be explained well, two CSI feedback algorithms based on depth expansion were proposed. The first one was approximate message delivery (AMP) algorithm based on learnable parameters. The learnable parameters in deep learning were used to replace the threshold value of the threshold function in the AMP algorithm and the parameters of the Onsager correction term. The nonlinear ability of threshold function in dealing with non-strict sparse data was enhanced. The other was the AMP algorithm based on convolutional network, which replaced the threshold function module with the convolutional residual learning module, and used the module to remove the Gaussian random noise generated by each iteration of the AMP algorithm. Simulation results show that the proposed two algorithms have better CSI feedback performance than AMP algorithm, and the AMP algorithm based on convolutional network has better CSI reconstruction performance than the representative method based on deep learning.

Keywords: CSI feedback, deep learning, deep unfolding, approximate message passing, learnable parameter, convolutional network

0 引言

大规模多输入多输出 (MIMO, multiple input multiple output) 技术凭借其高能量效率、高频谱效

率和大系统容量等优势成为下一代移动通信的关键技术之一^[1]。为了充分利用大规模 MIMO 技术带来的潜在性能增益, 基站需要获取足够精确的下行链路信道状态信息 (CSI, channel state information)^[2]。

收稿日期: 2022-08-30; 修回日期: 2022-11-17

基金项目: 重庆市自然科学基金资助项目 (No.cstc2019jcyj-msxmX0017)

Foundation Item: The Natural Science Foundation of Chongqing (No.cstc2019jcyj-msxmX0017)

在时分双工 (TDD, time division duplex) 大规模 MIMO 系统中, 可以利用上下行信道的互易性来获得 CSI。但是在频分双工 (FDD, frequency division duplexing) 大规模 MIMO 系统中, 用户必须先估计下行链路 CSI, 再通过上行链路反馈回基站^[3]。由于大规模 MIMO CSI 矩阵具备角度时延域稀疏特性, 压缩感知^[4]可以被应用于 CSI 反馈。虽然压缩感知理论与大规模 MIMO 的 CSI 反馈模型比较贴合, 但目前依然存在一些技术挑战, 因此研究人员开始关注深度学习在 CSI 领域的应用。文献[5]提出了第一个 FDD 大规模 MIMO 的 CSI 反馈网络 CsiNet, 它是一种通过引入自编码器结构^[6], 并且模拟压缩感知框架构建的高反馈精度的网络, 相比于传统的压缩感知方法, 有较好的反馈性能和更低的复杂度。为了进一步提升网络的恢复效果, 文献[7]在 CsiNet 的基础上提出了 CsiNet+, 该网络在用户设备处利用更大卷积核的卷积层增加 CSI 矩阵的感受野, 提取更多的深层次特征, 在基站处利用更多的卷积网络提高 CSI 重构精度。针对快时变 FDD 大规模 MIMO 系统中因多普勒频移导致的信道时间相关性使系统无法保证高可靠和低时延通信的问题, 文献[8]利用卷积神经网络 (CNN, convolutional neural network) 和批标准化网络对 CSI 矩阵中的噪声进行提取并且学习信道的空间结构, 通过注意力机制提取 CSI 矩阵间的时间相关性以提高 CSI 重构的精度。在高速移动环境下, 信道特征复杂多变, 同时存在加性噪声和非线性效应的影响, 文献[9]提出一种残差混合网络, 可消除高速移动场景加性噪声的影响, 学习并适应稀疏、双选衰落信道特征, 在高压缩率与低信噪比条件下依然具有较好的性能表现。针对 FDD 模式下 CSI 反馈方法复杂度高、精度低和开销大的问题, 文献[10]提出了一种轻量化低复杂度的反馈方法, 该方法在时间复杂度和空间复杂度上均有良好的表现。

值得注意的是, 大多数的深度学习方法需要依靠大量已分类标记的数据来进行神经网络的训练, 因此这种方法也被称为基于数据驱动的深度学习方法, 但该方法不具备可解释性。针对上述问题, 研究人员发现将传统的经典非线性迭代算法与深度学习的网络参数训练相结合, 可以减小对大量标记数据的强依赖性, 同时具备一定的理论保障和性能下限。这种通过结合经典非线性迭代算法的可解释性和深度学习高性能的方法被称作“深度展开”^[11]。文

献[12]提出一种多任务神经网络结构, 通过展开基于最小均方误差准则的两次迭代来恢复下行链路的 CSI 和上行用户的数据序列, 在减少上行链路带宽资源占用方面表现优异。文献[13]提出基于深度卷积神经网络的马尔可夫模型对前向 CSI 进行差分编码, 能有效提高 CSI 重构精度。文献[14]将快速迭代收缩阈值算法 (FISTA, fast iterative shrinkage thresholding algorithm) 展开为 FISTA-Net 并且作为大规模 MIMO 的 CSI 反馈算法, 获得了优异的性能。

针对基于深度学习的 CSI 反馈算法存在待训练参数过多、可解释性不强的问题, 本文提出了 2 种基于深度展开的 CSI 反馈算法, 以近似消息传递 (AMP, approximate message passing)^[15]作为深度展开的基准算法, 从 2 个视角分别展开得到了基于可学习参数的 AMP (LP-AMP, learnable parameters-AMP) 算法和基于卷积网络的 AMP (CNN-AMP, convolutional neural network-AMP) 算法。本文的主要贡献如下: 1) LP-AMP 算法通过改进 AMP 中的阈值函数和 Onsager 项来解决 CSI 矩阵非严格稀疏的问题; 2) 随着这种非严格稀疏性的加强, 在 LP-AMP 算法的基础上提出了利用卷积残差学习网络改进阈值函数模块的 CNN-AMP 算法, 与 LP-AMP 相比, CNN-AMP 在牺牲了一些复杂度的情况下, 获得了更加明显的重构效果; 3) 所提的基于深度展开的 CSI 反馈算法既具有压缩感知算法的可解释性理论保障, 又具有深度学习方法的高性能表现, 可以为 FDD 大规模 MIMO CSI 反馈领域的研究提供一些可行的思路。

1 系统模型

本文针对单小区大规模 MIMO 的通信系统, 基站配备有 N_t 根发射天线, 天线阵列方式采用均匀线性阵列, 用户端配备有单根接收天线。采用正交频分复用系统, 并有 N_s 个子载波, 因此接收向量 \mathbf{y} 可以建模为

$$\mathbf{y} = \tilde{\mathbf{H}}\mathbf{U}\mathbf{x} + \mathbf{n} \quad (1)$$

其中, \mathbf{y} 是维度为 $N_s \times 1$ 的复数接收向量; \mathbf{x} 是 $N_s \times 1$ 的复数发射向量; \mathbf{n} 是 $N_s \times 1$ 的加性高斯白噪声向量; $\tilde{\mathbf{H}}$ 是 $N_s \times N_t$ 的复数 CSI 信道矩阵, 其中行向量为每个子载波下的空域信道矢量, 列向量为频域信道矢量; \mathbf{U} 是 $N_t \times N_s$ 的复数预编码矩阵。如果以空频域矩阵作为反馈矩阵的基准, 那么 CSI 矩阵就是 $\tilde{\mathbf{H}}$,

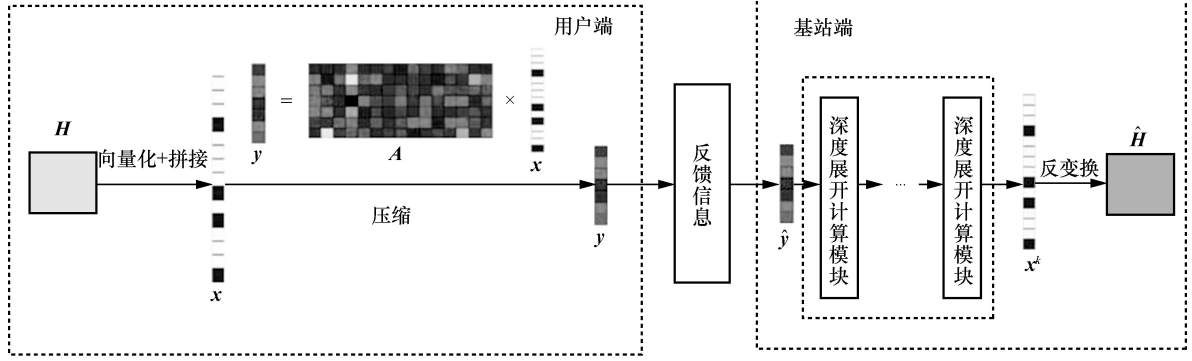


图 1 深度展开 CSI 反馈模型

此时的反馈总量是 $N_s N_t$ ，这会占据大量的系统资源，因此需要设计低复杂度高反馈精度的有限 CSI 反馈方法。

考虑到 CSI 矩阵具备角度时延域的稀疏特性，为了降低 CSI 的反馈开销，将角度和时延域的稀疏特性全部应用于空频域矩阵 \tilde{H} ，经过两次矩阵乘积的离散傅里叶变换（DFT, discrete Fourier transform），即可得到角度时延域稀疏 CSI 矩阵为

$$\tilde{H} = F_s \tilde{H} F_t^H \quad (2)$$

其中， F_s 和 F_t 分别是 $N_s \times N_s$ 和 $N_t \times N_t$ 维的 DFT 矩阵， \tilde{H} 是经两次 DFT 操作得到的角度时延域矩阵。经 DFT 之后的矩阵 \tilde{H} 在时延域上只有前 N'_s 行存在非零元素，其余行都为 $\mathbf{0}$ 向量^[5]，将 $\mathbf{0}$ 向量的行全部截断得到最终维度为 $N'_s \times N_t$ 的 CSI 矩阵 H ，此时的反馈总量从 $N_s N_t$ 缩减为了 $N'_s N_t$ ，反馈开销得到降低，不过 $N'_s N_t$ 的反馈总量对于通信系统而言依然是不小的开销。

将上述经 DFT 稀疏和截断之后的 CSI 矩阵 H 作为待反馈矩阵，深度展开 CSI 反馈模型如图 1 所示。首先，将用户端估计得到的 CSI 矩阵 H 经向量化和拼接得到长度为 $N = 2N'_s N_t$ 的实数稀疏向量 x ，在得到稀疏的待反馈向量 x 之后，测量矩阵使用独立同分布高斯随机矩阵 A 进行压缩。将矩阵的压缩和重构 2 个过程分别使用编码器和解码器进行描述，编码器在用户设备处，解码器在基站处。编码过程表示为

$$y = f_{\text{en}}(H) = A \left[\text{Re}(\text{vec}(H)); \text{Im}(\text{vec}(H)) \right] \quad (3)$$

其中， y 表示长度为 M 的测量向量， $f_{\text{en}}(\cdot)$ 表示编码器函数， $\text{Re}(\cdot)$ 表示取向量的实部， $\text{Im}(\cdot)$ 表示取向量的虚部， $\text{vec}(\cdot)$ 表示向量化操作。

y 通过反馈信道到达基站，为了便于区分，以 \hat{y} 表示基站端接收到的测量向量，基站通过深度展开的 AMP 算法恢复为第 k 次迭代的重构向量 x^k ，然后反变换回 CSI 稀疏矩阵 \hat{H} ，即可完成 CSI 的反馈流程。解码过程表示为

$$\hat{H} = f_{\text{de}}(\hat{y}) \quad (4)$$

其中， $f_{\text{de}}(\cdot)$ 表示解码器函数。最终的训练过程是将 $f_{\text{en}}(\cdot)$ 和 $f_{\text{de}}(\cdot)$ 联立在一起进行端到端的训练，这种端到端的训练过程可表示为

$$f_{\text{all}}(H_i, \Theta) = f_{\text{de}}(f_{\text{en}}(H_i, \Theta_{\text{en}}), \Theta_{\text{de}}) \quad (5)$$

其中， $f_{\text{all}}(\cdot)$ 是编码器和解码器联立后的变换函数， $\Theta = \{\Theta_{\text{en}}, \Theta_{\text{de}}\}$ 是网络的参数集合， Θ_{en} 是编码器参数， Θ_{de} 是解码器参数， H_i 是第 i 例训练样本。

2 基于 AMP 的深度展开 CSI 反馈

2.1 深度展开的原理

深度展开模型由多个计算模块组成，每个计算模块对应非线性迭代算法中的一次迭代。文献[16]指出，将深度学习方法与现有通信领域中的经典非线性迭代算法结合已成为近年来的研究热点，并且已经应用到预编码、信号检测和信道解码等处理中。深度展开模型架构如图 2 所示。

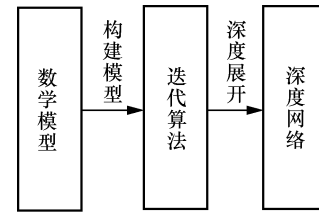


图 2 深度展开模型架构

深度展开模型包括 3 个部分，即数学模型、迭代算法和深度网络。在应用中将具体的实际问题抽

象为数学模型，根据相关的数学理论推导得到具体的迭代算法，将迭代算法每轮迭代的计算式展开为深度网络中的一层，利用可学习参数或神经网络替换原始迭代算法中每轮迭代中的超参数或某个计算模块。

2.2 AMP 算法

AMP 算法^[15]很好地继承了迭代阈值算法的优势，又融入了消息传递算法的精髓。假设初始化重构向量 $\mathbf{x}^0 = \mathbf{A}^T \mathbf{y}$ ，残差向量 $\mathbf{z}^{-1} = \mathbf{y}$ ，第 t ($t \geq 0$) 次迭代可表示为

$$\mathbf{x}^{t+1} = \eta_t(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t; \lambda_t) \quad (6)$$

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A} \mathbf{x}^t + b_t \mathbf{z}^{t-1} \quad (7)$$

其中， \mathbf{x}^t 是第 t 次迭代的长度为 N 的重构向量， \mathbf{z}^t 是第 t 次迭代的残差向量， \mathbf{A} 是 高斯随机矩阵， $b_t \mathbf{z}^{t-1}$ 是算法中的 Onsager 校正项。软阈值函数的常用形式为

$$\eta(u; \lambda) = \begin{cases} u - \lambda & , u \geq \lambda \\ u + \lambda & , u \leq -\lambda \\ 0 & \text{其他} \end{cases} \quad (8)$$

其中， u 为软阈值函数的输入。

根据文献[17]，Onsager 校正项中的 b_t 与阈值 λ_t 为

$$b_t = \frac{1}{M} \|\mathbf{x}^t\|_0 \quad (9)$$

$$\lambda_t = \frac{\varepsilon}{\sqrt{M}} \|\mathbf{z}^t\|_2 \quad (10)$$

其中， ε 是一个与稀疏度和测量向量长度 M 有关的可调数值。AMP 算法与 FISTA 相似，不同之处在于 AMP 算法在残差向量更新时增加了 Onsager 校正项，并且迭代阈值在每轮迭代中都会通过残差向量和测量向量长度 M 进行调整。AMP 计算模块如图 3 所示。

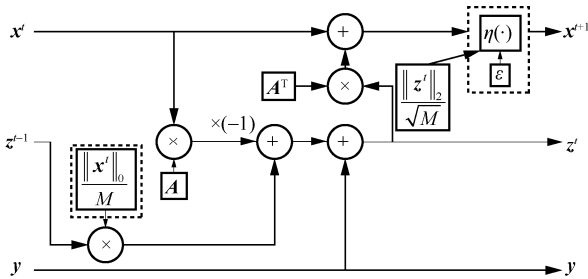


图 3 AMP 计算模块

图 3 展示了 AMP 算法迭代式中的每一个计算模块，AMP 算法的流程如算法 1 所示。

算法 1 AMP 算法

输入 测量向量 \mathbf{y} ，高斯随机矩阵 \mathbf{A}

输出 重构向量 \mathbf{x}^{t+1}

初始化 重构向量 $\mathbf{x}^0 = \mathbf{A}^T \mathbf{y}$ ，残差向量 $\mathbf{z}^{-1} = \mathbf{y}$ ，迭代次数 $t=1$

步骤 1 由测量向量 \mathbf{y} 、高斯随机矩阵 \mathbf{A} 、当次迭代的 \mathbf{x}^t 、Onsager 校正项 $b_t \mathbf{z}^{t-1}$ 计算出当次迭代的 $\mathbf{z}^t = \mathbf{y} - \mathbf{A} \mathbf{x}^t + b_t \mathbf{z}^{t-1}$ ；

步骤 2 如果 $\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t$ 的值大于或等于阈值 λ_t ，则得到 $\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t - \lambda_t$ ，如果 $\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t$ 的值小于或等于 $-\lambda_t$ ，则得到 $\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t + \lambda_t$ ，否则得到 $\mathbf{x}^{t+1} = 0$ ；

步骤 3 $t=t+1$ ，若 $t < N$ ，返回步骤 1，若 $t = N$ ，迭代结束；

2.3 基于可学习参数的 AMP 算法

本节对基于可学习参数的 AMP 反馈算法 LP-AMP 的展开结构进行描述。具体而言，通过引入深度学习中梯度下降法更新网络参数，优化 AMP 算法中的迭代更新参数，将算法的每轮迭代展开为深度网络中的一层，再将多个这样的层堆叠，在模型训练的过程中逐渐优化每层中不同的可学习参数，使模型以更少的迭代层数更快地收敛。LP-AMP 的中间展开计算模块如图 4 所示。

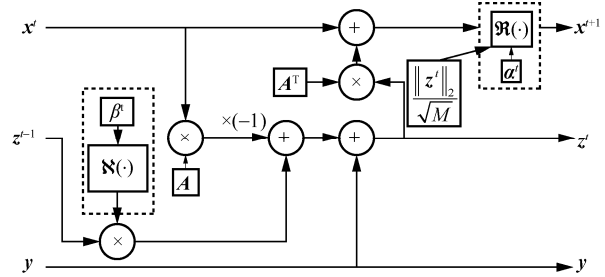


图 4 LP-AMP 的中间展开计算模块

输入层对进入迭代中的 \mathbf{x}^t 和 \mathbf{z}^t 进行初始化操作，输出层将迭代至最后的 \mathbf{x}^k 作为 LP-AMP 的最后输出，中间展开计算模块的 $\mathfrak{R}(\cdot)$ 和 $\mathfrak{S}(\cdot)$ 函数是引入可学习参数 α^t 向量和 β^t 参数的 2 个可学习模块。

LP-AMP 算法的初始化和中间展开式可以表示为

$$\mathbf{x}^0 = \mathbf{Q} \mathbf{y} \quad (11)$$

$$\mathbf{z}^{-1} = \mathbf{y} \quad (12)$$

$$\mathbf{x}^{t+1} = \mathfrak{R} \left(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t; \frac{\|\mathbf{z}^t\|_2}{\sqrt{M}} \alpha^t \right) \quad (13)$$

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + \frac{1}{M} \mathfrak{S}(\mathbf{x}^t; \beta^t) \mathbf{z}^{t-1} \quad (14)$$

其中, \mathbf{x}^0 的初始化是通过将采样矩阵的转置 \mathbf{A}^T 作为可学习矩阵 \mathbf{Q} 的初始化, 在不断训练的过程中, 调整矩阵 \mathbf{Q} 的分布, 增强初始重构效果; α^t 是第 t 层展开中的可学习向量; β^t 是第 t 层展开中的可学习参数。对于展开式中的 $\mathfrak{R}(\cdot)$ 和 $\mathfrak{S}(\cdot)$ 模块, $\mathfrak{R}(\cdot)$ 模块对应原始 AMP 算法中的阈值函数 $\eta(\cdot)$ 部分, 与阈值函数有所不同; $\mathfrak{S}(\cdot)$ 模块则对应于 Onsager 项部分中的对 \mathbf{x}^t 的 0 范数计算过程。

阈值函数 $\eta(\cdot)$ 和 $\mathfrak{R}(\cdot)$ 模块的对比如图 5 所示。

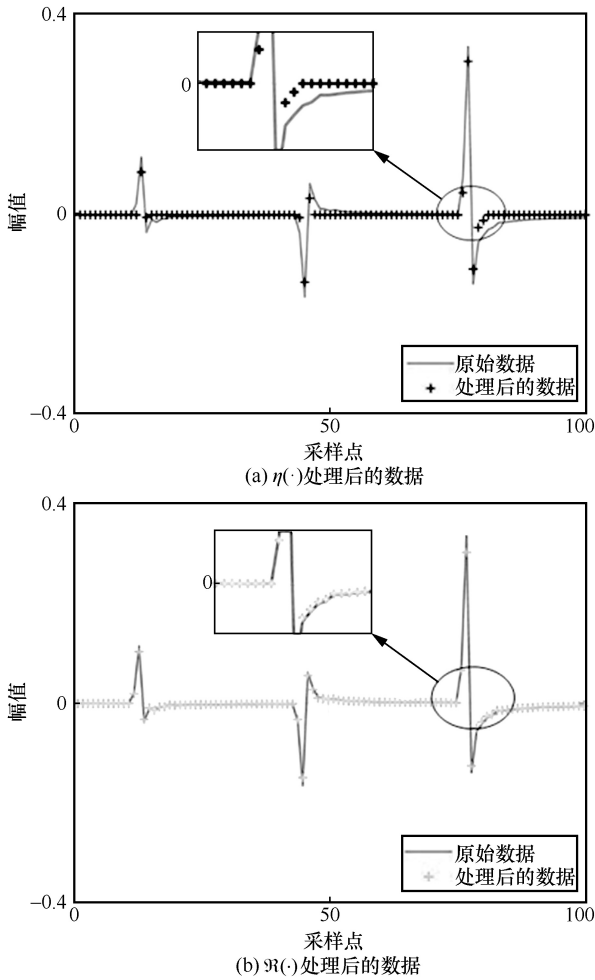


图 5 阈值函数 $\eta(\cdot)$ 和 $\mathfrak{R}(\cdot)$ 模块的对比

从图 5 可以看出, 原始的 CSI 数据并非严格稀疏, 大多数的数据非常接近 0, 但是并不为 0。阈值函数 $\eta(\cdot)$ 则在一次更新迭代的情况下将绝对值小于阈值 λ_t 的参数严格置 0, 这破坏了原始信号的细节 (即小于阈值但不为 0 的点在重构信号中被删除)。本文将原始的阈值函数 $\eta(\cdot)$ 替换成 $\mathfrak{R}(\cdot)$ 计算模块,

是为了加强阈值函数在应对非严格稀疏数据时的非线性表征能力。与 $\eta(\cdot)$ 阈值函数不同的是, $\mathfrak{R}(\cdot)$ 模块引入可学习向量 α^t 来提升整体的变换效果, 计算式为

$$\mathfrak{R}\left(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t; \frac{\|\mathbf{z}^t\|_2}{\sqrt{M}} \alpha^t\right) = \left(\left|\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t\right| - \frac{\|\mathbf{z}^t\|_2}{\sqrt{M}} \alpha^t\right) \text{sign}(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t) \quad (15)$$

其中, $\alpha^t \in \mathbb{R}^N$ 是长度为 N 的可学习参数向量, 它的长度与重构向量 \mathbf{x}^t 的长度相同, 即 $N = 2N_s N_t$ 。 $\mathfrak{R}(\cdot)$ 模块的计算式使不会有小于阈值的点被严格置 0, 并且每个数据点都有一个可学习参数进行调整。

$\mathfrak{S}(\cdot)$ 模块对应于 AMP 算法中 Onsager 中求解 \mathbf{x}^t 的 0 范数部分, \mathbf{x}^t 的 0 范数是指向量 \mathbf{x}^t 中的非零元素的数目。但是通过 $\mathfrak{R}(\cdot)$ 模块处理后的数据几乎不存在严格为 0 的数据, 因此 $\mathfrak{S}(\cdot)$ 模块中对 0 范数的处理同样需要通过可学习参数来模拟。具体而言, 将可学习参数 β^t 作为计算非零元素数目的阈值, 超过 β^t 阈值的数据点被判定为非零元素。通过后续的仿真实验可以看到, 引入可学习参数 β^t 可以减少算法的总迭代次数并提高重构精度。另外, 值得注意的是, 采样矩阵 \mathbf{A} 是服从高斯分布的随机矩阵。然而, 已有研究证明, 设计合理的采样矩阵可以提高 CSI 矩阵重构的精度^[18]。在 LP-AMP 中也不例外, 为了提高重构精度, 利用经训练过后的 \mathbf{A} 进行数据的压缩, \mathbf{Q} 进行数据的初始重构。LP-AMP 算法的流程如算法 2 所示。

算法 2 LP-AMP 算法

输入 测量向量 \mathbf{y} , 训练后的矩阵 \mathbf{A}

输出 重构向量 \mathbf{x}^{t+1}

初始化 重构向量 $\mathbf{x}^0 = \mathbf{Q}\mathbf{y}$, 训练后的矩阵 \mathbf{Q} 的初始化为 \mathbf{A}^T , 残差向量 $\mathbf{z}^{-1} = \mathbf{y}$, 迭代次数 $t=1$

步骤 1 由测量向量 \mathbf{y} 、训练后的矩阵 \mathbf{A} 、当次迭代的 \mathbf{x}^t 、当前展开计算模块中的可学习参数 β^t 、前一次迭代的 \mathbf{z}^{t-1} 计算出当次迭代的 $\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + M^{-1} \mathfrak{S}(\mathbf{x}^t; \beta^t) \mathbf{z}^{t-1}$;

步骤 2 引入可学习参数 α^t 向量, 利用当前展开计算模块中的 $\mathfrak{R}(\cdot)$ 计算得到 $\mathbf{x}^{t+1} = \left(\left|\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t\right| - M^{-\frac{1}{2}} \|\mathbf{z}^t\|_2 \alpha^t\right) \text{sign}(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t)$;

步骤 3 $t=t+1$, 若 $t < N$, 返回步骤 1, 若 $t=N$, 迭代结束;

2.4 基于卷积网络的 AMP 算法

LP-AMP 的 $\mathfrak{R}(\cdot)$ 阈值模块依然面临一个重要的问题, 即虽然 LP-AMP 通过可学习参数作为每层的阈值向量, 但是在训练之后会使 $\mathfrak{R}(\cdot)$ 模块中的阈值向量 α' 成为固定值, 这会使个别不符合总体样本特征的数据重构效果变差, 进而影响算法的整体泛化能力。在 LP-AMP 的整个流程中, $\mathfrak{R}(\cdot)$ 模块几乎承担着大部分的非线性计算过程, 而深度学习中 CNN 在应对非线性表征时, 往往有着更好的性能。因此本节利用卷积残差学习模块替换 LP-AMP 中的阈值模块 $\mathfrak{R}(\cdot)$, 其余部分不做任何改动。为了方便后续的描述, 将基于卷积残差学习网络的 AMP 反馈算法命名为 CNN-AMP。从理论推导角度而言, LP-AMP 中 $\mathfrak{R}(\cdot)$ 的作用是去除数据中的高斯白噪声影响。具体的推导过程如下。

根据 $y = Ax$ 可得

$$\begin{aligned} A^T y &= A^T Ax = \\ A^T Ax - x + x &= x + (A^T A - I)x \end{aligned} \quad (16)$$

其中, $(A^T A - I)x$ 在文献[15]中被证明服从高斯分布, 且方差为 $\frac{1}{M} \|x\|_2^2$, I 是单位矩阵。简单来说, 初始化的 x^0 是带有高斯噪声的重构向量, AMP 的迭代过程是为了逐渐减小噪声的影响, 令 $(A^T A - I) = P$, 可得

$$\begin{aligned} A^T z^0 + x^0 &= A^T (y - Ax^0) + x^0 = \\ A^T (Ax - Ax^0) + x^0 &= A^T A(x - x^0) + \\ x^0 &= x + P(x - x^0) \end{aligned} \quad (17)$$

其中, $P(x - x^0)$ 同样服从高斯分布, 且方差为 $\frac{1}{M} \|x - x^0\|_2^2$ 。将式(17)推广至第 t 次迭代可得

$$\begin{aligned} A^T z^t + x^t &= A^T A(x - x^t) + x^t = \\ x + P(x - x^t) \end{aligned} \quad (18)$$

其中, $P(x - x^t)$ 服从高斯分布, 且方差为 $\frac{1}{M} \|x - x^t\|_2^2$ 。到这里可以发现, 随着迭代次数的增加, x^t 与 x 逐渐接近, 方差 $\frac{1}{M} \|x - x^t\|_2^2$ 逐渐减小, 重构过程逐渐收敛。

根据上述推导, AMP 算法中 $\eta(\cdot)$ 阈值函数的作用是去除每轮迭代重构信号中的高斯白噪声。

CNN-AMP 的中间展开计算模块如图 6 所示, 其与 LP-AMP 的不同之处仅在于将 $\mathfrak{R}(\cdot)$ 模块替换为了 $\wp(\cdot)$ 模块, $\wp(\cdot)$ 模块由 3 个深度可分离卷积层^[19]构成, 卷积核的大小为 3×3 , 每个卷积层之间通过 ReLU 激活函数相连。值得强调的是, $\wp(\cdot)$ 模块利用残差学习^[20]的模式, 将网络的输入减去网络的输出得到最终结果, 即 $\wp(\cdot)$ 模块将学习到每轮重构向量的高斯白噪声, 最后通过输入减去噪声得到该展开层信号的重构。

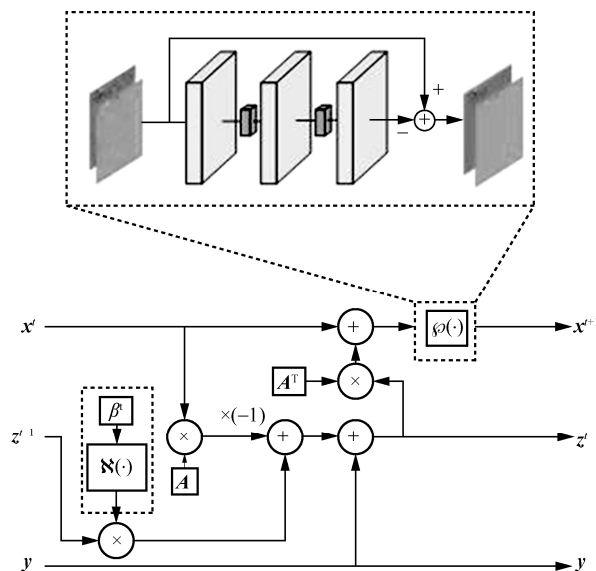


图 6 CNN-AMP 的中间展开计算模块

根据上述对 CNN-AMP 的描述, 它的迭代展开式可以写为

$$x^{t+1} = \wp(x^t + A^T z^t) \quad (19)$$

$$z^t = y - Ax^t + \frac{1}{M} \mathfrak{N}(x^t; \beta^t) z^{t-1} \quad (20)$$

其中, CNN-AMP 中的 $\mathfrak{N}(x^t; \beta^t)$ 模块与 LP-AMP 保持一致, 依然利用学习参数 β^t 作为模拟求解 x^t 的 0 范数阈值。CNN-AMP 的输入层的初始化过程以及输出层也与 LP-AMP 保持一致, 为了提高重构精度, 采样矩阵 A 与初始重构矩阵 Q 同样加入网络参数的训练。CNN-AMP 算法的流程如算法 3 所示。

算法 3 CNN-AMP 算法

输入 测量向量 y , 训练后的矩阵 A

输出 重构向量 x^{t+1}

初始化 重构向量 $x^0 = Qy$, 训练后的矩阵 Q 的初始化为 A^T , 残差向量 $z^{-1} = y$, 迭代次数 $t=1$

步骤 1 由测量向量 \mathbf{y} 、训练后的矩阵 \mathbf{A} 、当次迭代的 \mathbf{x}^t 、当前展开计算模块中的可学习参数 β^t 、前一次迭代的 \mathbf{z}^{t-1} 计算出当次迭代的 $\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + M^{-1}\mathfrak{S}(\mathbf{x}^t; \beta^t)\mathbf{z}^{t-1}$;

步骤 2 引入残差学习模块 $\wp(\cdot)$ ，利用当前展开计算模块中的 $\wp(\cdot)$ 计算得到 $\mathbf{x}^{t+1} = \wp(\mathbf{x}^t + \mathbf{A}^T \mathbf{z}^t)$;

步骤 3 $t=t+1$ ，若 $t < N$ ，返回步骤 1，若 $t=N$ ，迭代结束；

3 模型训练与复杂度分析

3.1 模型训练

对比 LP-AMP、CNN-AMP 和 AMP 的展开或迭代式可以发现，若去掉 LP-AMP 和 CNN-AMP 中的 $\mathfrak{R}(\cdot)$ 、 $\mathfrak{S}(\cdot)$ 和 $\wp(\cdot)$ 模块，将其更改为原始阈值函数 $\eta(\cdot)$ 和常规的 0 范数计算，LP-AMP 和 CNN-AMP 就会退化为 AMP 算法，也就是说，LP-AMP 和 CNN-AMP 的性能下限是可以预知的，即存在理论前提和性能保障。但是基于数据驱动的深度学习方法的性能则无法提前预知，在进行训练之前，无法保证得到很好的效果。为了方便后续对比实验的分析，这里 k_1 层展开的 LP-AMP 和 k_2 层展开的 CNN-AMP 被命名为 LP-AMP- k_1 和 CNN-AMP- k_2 。产生数据集的信道模型是基于空间几何的随机信道模型 COST2100^[21]，其在 COST259 模型和 COST273 模型的基础上进行构建，可以在时间、频率和空间上重现 MIMO 信道的随机属性。产生的数据集分别为 100 000 训练集、30 000 验证集和 20 000 测试集。采用端到端的训练方式，训练的损失函数采用均方误差 (MSE, mean square error)，它是一种计算预测值和真实值之间误差的函数，具体的表达式为

$$L(\Theta) = \frac{1}{T} \sum_{i=1}^T (f_{\text{all}}(\mathbf{H}_i; \Theta) - \mathbf{H}_i)^2 \quad (21)$$

其中， \mathbf{H}_i 为原始输入的第 i 例样本，矩阵的维度是 32×32 ， Θ 为网络中的所有参数，包括 \mathbf{A} 、 \mathbf{Q} 、 α^t 、 β^t 等， $f_{\text{all}}(\cdot)$ 为整个网络的计算式， $f_{\text{all}}(\mathbf{H}_i; \Theta)$ 为恢复的第 i 例样本， T 为总的训练样本数量。优化器算法为自适应矩估计^[22]优化器，它是随机梯度下降算法的一种变体。采样矩阵 \mathbf{A} 使用服从高斯分布的随机矩阵初始化填充，第 t 层可训练向量 α^t 使用 0.1 初始化填充，第 t 层 β^t 也初始化为 0.1。

为了得到理想效果并避免梯度消失的问题，初始学习率设置为 0.001，衰减率设置为 0.9。使用的训练设备配置为 NVIDIA GeForce RTX 2080 Ti 显卡，Intel Xeon E5-2678 v3 处理器，64 GB 内存。通过实验发现，LP-AMP- k_1 的收敛速度比 CNN-AMP- k_2 快得多。当压缩率 $\gamma = \frac{1}{16}$ ，批量数据大小为 200， k_1 和 k_2 分别取 3 和 7 时，LP-AMP- k_1 和 CNN-AMP- k_2 的 MSE 如表 1 所示，采用 Python 平台的深度学习框架 Pytorch 搭建整体结构。

表 1 LP-AMP- k_1 和 CNN-AMP- k_2 的 MSE

训练轮次	LP-AMP-3	LP-AMP-7	CNN-AMP-3	CNN-AMP-7
10	5.930×10^{-5}	5.931×10^{-5}	2.008×10^{-5}	1.015×10^{-5}
30	5.929×10^{-5}	5.716×10^{-5}	1.271×10^{-5}	7.029×10^{-6}
50	5.928×10^{-5}	5.599×10^{-5}	1.162×10^{-5}	5.993×10^{-6}
90	5.933×10^{-5}	5.580×10^{-5}	1.112×10^{-5}	5.309×10^{-6}
110	5.936×10^{-5}	5.553×10^{-5}	1.089×10^{-5}	5.032×10^{-6}
200	5.933×10^{-5}	5.561×10^{-5}	1.078×10^{-5}	5.118×10^{-6}
300	5.931×10^{-5}	5.559×10^{-5}	1.074×10^{-5}	5.125×10^{-6}

除此之外，还需要确定 AMP 算法的收敛迭代次数，及 LP-AMP- k_1 和 CNN-AMP- k_2 的最佳展开层数。当室内场景下压缩率 $\gamma = \frac{1}{16}$ ，LP-AMP- k_1 和 CNN-AMP- k_2 在验证集上的 MSE 如图 7 所示。从图 7 可以看出，当 $k_1=1, 3, 5, 7, 9$ 时，LP-AMP- k_1 整体损失呈下降趋势，但是损失整体相差并不大，随着 k_1 的增大，最终收敛的迭代次数开始增加， $k_1=3$ 和 $k_1=5$ 时的收敛速度几乎相同， $k_1=5$ 时的整体损失仅略低于 $k_1=3$ 。综合复杂度和性能平衡之下，设置 $k_1=3$ 。另外，CNN-AMP- k_2 整体的损失比 LP-AMP- k_1 低一个数量级，这也是在牺牲一些复杂度之下换取的性能优势。随着层数的增加，CNN-AMP- k_2 损失同样呈下降趋势， $k_2=5$ 和 $k_2=7$ 时的损失差距不大，为了平衡复杂度和性能，设置 $k_2=5$ 。

AMP 的迭代次数 q 也是影响整体复杂度的关键参数，MSE 随迭代次数的变化如图 8 所示。无论在室内或室外场景，AMP 算法信道矩阵的非严格稀疏性导致算法在验证集上整体收敛于迭代次数 50~100，以下的计算选取 $q=100$ 。相比于 LP-AMP-3，AMP 的乘法次数高一个数量级，与 CNN-AMP-5 较接近。

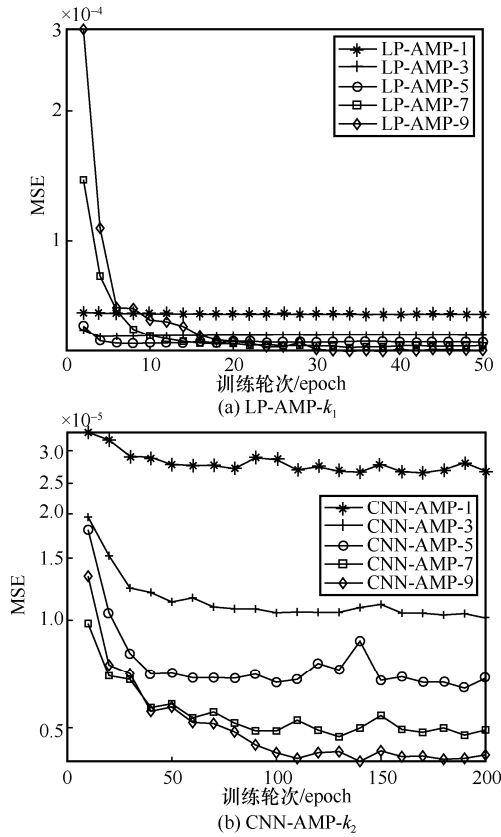


图 7 LP-AMP- k_1 和 CNN-AMP- k_2 在验证集上的 MSE

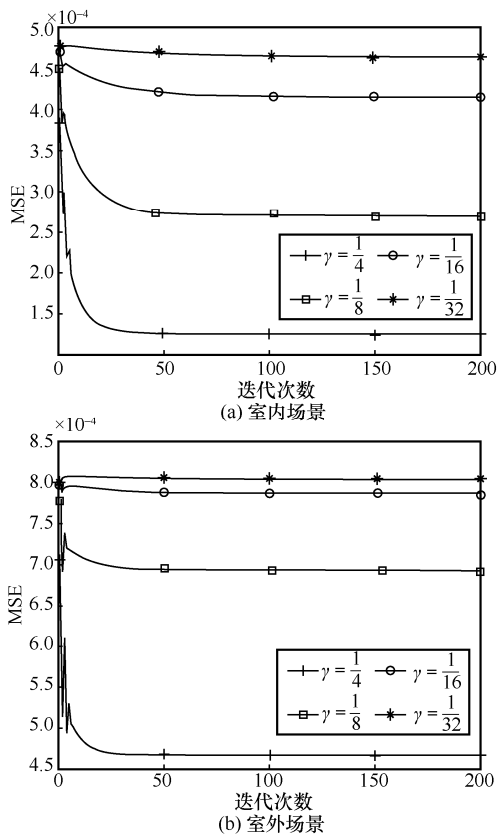


图 8 MSE 随迭代次数的变化

3.2 复杂度分析

本节将通过对比 CsiNet、CsiNetPlus、LP-AMP-3、CNN-AMP-5 和 AMP 方法来说明深度展开方法的优势和缺陷，通过分析网络的参数量和实数乘法次数来反映复杂度。对于网络参数，AMP 作为传统的压缩感知算法并不存在网络参数，但是 LP-AMP-3 和 CNN-AMP-5 是 AMP 的深度展开算法，引入了可学习参数和残差学习模块，需要计算整体的参数量。

参数的计算将 $M=512$ 、 256 、 128 、 64 和 $N=2048$ 代入计算，并且忽略所有结构的偏置项。不同算法编码器和解码器的参数量如表 2 所示。

LP-AMP-3 和 CNN-AMP-5 算法编码器部分的参数量与 CsiNet 和 CsiNetPlus 几乎相同，这是因为 LP-AMP-3 和 CNN-AMP-5 都将采样矩阵 A 作为可训练参数，这与全连接神经网络的压缩过程在参数量的计算上是等价的。

在解码器部分，LP-AMP-3 和 CNN-AMP-5 比 CsiNet 和 CsiNetPlus 的参数量略大，不过总体来说， $M \times N$ 依然是参数量计算的主体部分，数量级依然相同。LP-AMP-3 存在可训练矩阵 Q 以及 $\mathfrak{R}(\cdot)$ 、 $\mathfrak{S}(\cdot)$ 模块中的向量 α' 和参数 β' ，CNN-AMP-5 与 LP-AMP-3 同样存在可训练矩阵 Q ，不同之处在于， $\mathfrak{R}(\cdot)$ 模块被替换为 $\varphi(\cdot)$ 模块，当层数越大时，参数量相应也就越大。

不同算法的乘法次数如表 3 所示。AMP、LP-AMP-3 和 CNN-AMP-5 在编码器部分的乘法次数一样，这是因为在压缩感知算法的框架中，编码器部分都是采样矩阵 A 与待压缩向量 x 的乘积，而所提算法和 CsiNet 和 CsiNetPlus 的乘法次数几乎相同，这是因为它们的计算量主要来源于 $M \times N$ 。在解码器部分，LP-AMP-3 在压缩率为 $\frac{1}{4}$ 时比 CsiNet 和

CsiNetPlus 的乘法次数多，在其他压缩率时则相反，这是因为所提算法的计算量不仅与展开层数有关，并且受 $M \times N$ 的影响较大。不管在哪种压缩率下，CNN-AMP-5 的乘法次数都高于 CsiNet 和 CsiNetPlus，这主要是受 $\varphi(\cdot)$ 模块的影响。和 AMP 算法相比，LP-AMP-3 和 CNN-AMP-5 的复杂度较低，这是因为它们的展开层数远小于 AMP 的总迭代次数 100。另外，CNN-AMP-5 的复杂度比 LP-AMP-3 高，原因在于 $\varphi(\cdot)$ 模块使用了连续的卷积层。

表 2 不同算法编码器和解码器的参数量

压缩率	编/解码器	参数量/个			
		CsiNet	CsiNetPlus	LP-AMP-3	CNN-AMP-5
$\frac{1}{4}$	编码器	1 048 612	1 049 264	1 048 576	1 048 576
	解码器	1 051 780	1 052 486	1 054 723	1 052 901
$\frac{1}{8}$	编码器	524 324	524 986	524 288	524 288
	解码器	527 492	527 548	530 435	528 613
$\frac{1}{16}$	编码器	262 180	262 424	262 144	262 144
	解码器	265 384	265 846	268 291	266 469
$\frac{1}{32}$	编码器	131 108	131 342	131 072	131 072
	解码器	134 312	134 462	137 219	135 397

表 3 不同算法的乘法次数

压缩率	编/解码器	乘法次数/次				
		AMP	CsiNet	CsiNetPlus	LP-AMP-3	CNN-AMP-5
$\frac{1}{4}$	编码器	1 048 576	1 085 440	1 089 642	1 048 576	1 048 576
	解码器	210 917 376	4 329 472	4 338 284	7 270 912	15 960 576
$\frac{1}{8}$	编码器	524 288	561 152	567 254	524 288	524 288
	解码器	105 458 688	3 805 184	3 823 128	3 672 320	10 192 128
$\frac{1}{16}$	编码器	262 144	299 008	308 238	262 144	262 144
	解码器	52 729 344	3 543 040	3 583 462	1 836 160	7 307 904
$\frac{1}{32}$	编码器	131 072	167 936	179 928	131 072	131 072
	解码器	26 364 672	3 411 968	3 472 842	918 080	5 865 792

4 仿真分析

本节将对 LP-AMP-3、CNN-AMP-5、CsiNet、CsiNetPlus 和 AMP 算法的 CSI 反馈性能。经两次 DFT 得到角度时延域矩阵 \mathbf{H} , 保留该矩阵的前 $N_s' = 32$ 行, 因此复数矩阵 \mathbf{H} 的维度为 32×32 , 再将复数矩阵改为实虚部双通道矩阵。压缩率 $\gamma = \frac{M}{N}$, $M=512、256、128、64$, $N=2\ 048$ 。在 MATLAB 平台进行仿真, 设备配置与第 3 节相同, 仿真系统主要参数如表 4 所示, 其他参数都遵循文献[21]中的默认设置。

4.1 归一化均方误差

将离线训练好的模型应用于室内室外场景下的测试集, 并通过计算原始矩阵与重构矩阵之间的归一化均方误差 (NMSE, normalized mean squared error) 对模型的性能进行量化评估。NMSE 的计算式为

$$\text{NMSE} = \text{E} \left(\frac{\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_2^2}{\|\mathbf{H}_i\|_2^2} \right) \quad (22)$$

其中, \mathbf{H}_i 和 $\hat{\mathbf{H}}_i$ 分别表示原始信道数据和恢复信道数据。不同方法在室内和室外场景下的 NMSE 如图 9 所示。

表 4 仿真系统主要参数

仿真参数	设置
双工模式	FDD
发射天线数	32
接收天线数	1
子载波数	1 024
信道模型	COST2100
调制方式	正交相移键控
室内场景频段/GHz	5.3
室外场景频段/MHz	300
系统带宽/MHz	20

室内场景下, LP-AMP-3 在 $\frac{1}{4}$ 和 $\frac{1}{8}$ 的高压缩率时 NMSE 分别为 -29.26 dB 和 -15.74 dB, 在 $\frac{1}{16}$ 和 $\frac{1}{32}$ 的低压缩率时 NMSE 分别为 -9.12 dB 和 -4.83 dB, 与 CsiNet 和 CsiNetPlus 算法比较接近。室外场景下, LP-AMP-3 近乎退化为 AMP 算法, 这是由于室外环境的多变使信道的稀疏性变得非常差, 而 LP-AMP-3 中的阈值函数是 AMP 阈值函数的改进, 本质上并没有脱离 AMP 算法的框架, 因此在室外场景下, LP-AMP-3 的 NMSE 性能与 AMP 较为相似。对于 CNN-AMP-5 来说, 无论是室内场景还是室外场景, 都具有明显的 NMSE 优势。在室内场景下, CNN-AMP-5 在压缩率为 $\frac{1}{32}$ 时有着超出 CsiNet 算法 7.63 dB 和超出 CsiNetPlus 算法 5.86 dB 的优势; 在室外场景下, CNN-AMP-5 在高压缩率下的优势更加明显。

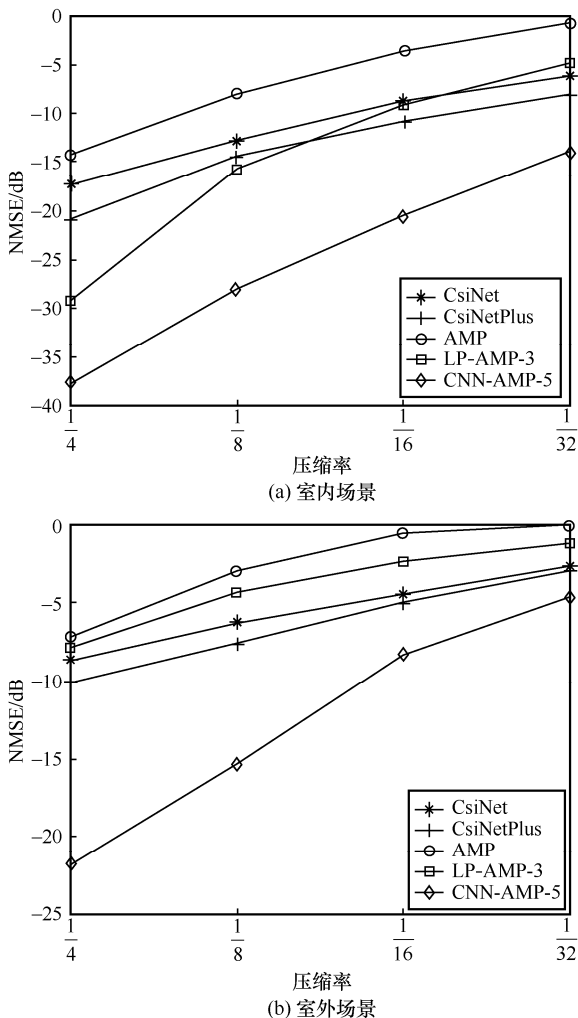


图 9 不同方法在室内和室外场景下的 NMSE

4.2 相似度

除 NMSE 之外, 余弦相似度 ρ 同样可以用来对比不同算法之间 CSI 矩阵的重构精度。该指标可以体现重构的 CSI 矩阵与原始 CSI 矩阵之间的相似程度, 其计算式为

$$\rho = \frac{|\mathbf{H}_{\text{vec}} \hat{\mathbf{H}}_{\text{vec}}|}{\|\mathbf{H}_{\text{vec}}\|_2 \|\hat{\mathbf{H}}_{\text{vec}}\|_2} \quad (23)$$

其中, \mathbf{H}_{vec} 、 $\hat{\mathbf{H}}_{\text{vec}}$ 分别表示向量化后的 CSI 信道矩阵和重构的 CSI 信道矩阵, ρ 值越大, 说明两者越相似, 重构精度越高。不同方法在室内和室外场景下的余弦相似度如图 10 所示。

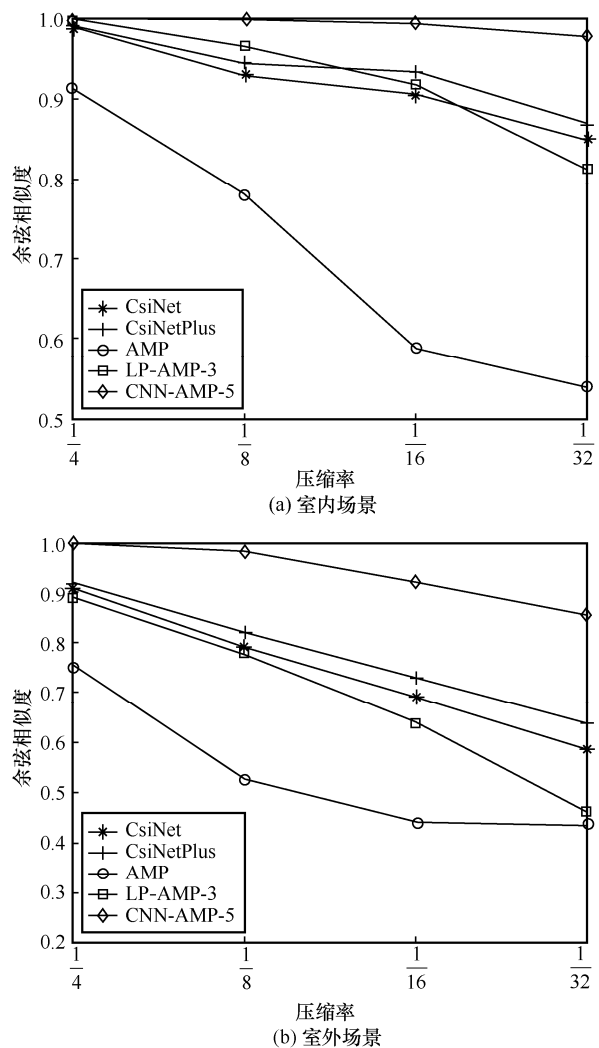


图 10 不同方法在室内和室外场景下的余弦相似度

由图 10 可知, 经典压缩感知的 AMP 算法在不同场景、不同压缩率下的 ρ 均低于其他几种算法, 即 CSI 矩阵的重构相似度均低于其他算法。在室内场景下,

AMP 算法在压缩率为 $\frac{1}{16}$ 时 ρ 明显下降, 而在室外场景下, AMP 在各压缩率下的 ρ 都不够理想, 已无法满足大规模 MIMO 系统 CSI 反馈的精度要求。LP-AMP-3 算法在室内场景的不同压缩率下的 ρ 与 CsiNet 和 CsiNetPlus 算法比较接近, 在室外场景的不同压缩率下的 ρ 略低于 CsiNet 和 CsiNetPlus 算法。CNN-AMP-5 算法不仅在室内场景的不同压缩率下保持较高的 ρ , 在室外场景下也有着优异的表现, 可以满足大规模 MIMO 系统 CSI 反馈的精度要求。

5 结束语

本文研究了基于深度展开的大规模 MIMO CSI 反馈算法。在 AMP 算法迭代过程的基础上, 从 2 个角度将其深度展开, 提出了基于可学习参数的 AMP 深度展开算法 LP-AMP 和基于卷积残差学习模块的 AMP 深度展开算法 CNN-AMP。针对信道矩阵的非严格稀疏性的问题, LP-AMP 通过深度学习的方法改进了阈值函数, 增强了 AMP 算法应对室内场景的性能, 降低了算法的复杂度。在室外场景下, 随着信道矩阵的剧烈变化, 非严格稀疏性问题逐渐增大, 针对这个问题, 本文又提出 CNN-AMP 算法, 将 LP-AMP 算法中改进的阈值函数模块替换为卷积残差学习模块, 该模块可以消除 AMP 算法在每次迭代更新后重构向量中的高斯随机噪声。因此无论是在室内场景下还是在室外场景下, 与代表性算法相比, 所提算法都有明显的优势。本文提出的基于深度展开的 CSI 反馈算法既具有压缩感知算法的可解释性理论保障, 又具有深度学习方法的高性能表现, 可以为大规模 MIMO CSI 反馈提供一些参考, 未来还可以从降低复杂度方面对算法进行改进。

参考文献:

- [1] LI C G, LIU P, ZOU C, et al. Spectral-efficient cellular communications with coexistent one- and two-hop transmissions[J]. IEEE Transactions on Vehicular Technology, 2016, 65(8): 6765-6772.
- [2] BARRIAC G, MADHOW U. Space-time communication for OFDM with implicit channel feedback[J]. IEEE Transactions on Information Theory, 2004, 50(12):3111-3129.
- [3] RAO X B, LAU V K N. Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems[J]. IEEE Transactions on Signal Processing, 2014, 62(12): 3261-3271.
- [4] LIANG P Z, FAN J C, SHEN W H, et al. Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 9217-9222.
- [5] WEN C K, SHIH W T, JIN S. Deep learning for massive MIMO CSI feedback[J]. IEEE Wireless Communications Letters, 2018, 7(5): 748-751.
- [6] WANG Y, YAO H, ZHAO S. Auto-encoder based dimensionality reduction[J]. Neurocomputing, 2016, 184: 232-242.
- [7] GUO J J, WEN C K, JIN S, et al. Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: design, simulation, and analysis[J]. IEEE Transactions on Wireless Communications, 2020, 19(4): 2827-2840.
- [8] 廖勇, 王帅, 孙宁. 快时变 FDD 大规模 MIMO 系统智能 CSI 反馈方法[J]. 通信学报, 2021, 42(7): 211-219.
LIAO Y, WANG S, SUN N. Intelligent CSI feedback method for fast time-varying FDD massive MIMO system[J]. Journal on Communications, 2021, 42(7): 211-219.
- [9] 廖勇, 王世义. 高速移动环境下基于 RM-Net 的大规模 MIMO CSI 反馈算法[J]. 通信学报, 2022, 43(5): 166-176.
LIAO Y, WANG S Y. CSI feedback algorithm based on RM-Net for massive MIMO systems in high-speed mobile environment[J]. Journal on Communications, 2022, 43(5): 166-176.
- [10] 廖勇, 李玉杰. 一种轻量化低复杂度的 FDD 大规模 MIMO 系统 CSI 反馈方法[J]. 电子学报, 2022, 50(5): 1211-1217.
LIAO Y, LI Y J. Lightweight and low complexity CSI feedback method for FDD massive MIMO systems[J]. Acta Electronica Sinica, 2022, 50(5): 1211-1217.
- [11] JAGANNATH A, JAGANNATH J, MELODIA T. Redefining wireless communication for 6G: signal processing meets deep learning with deep unfolding[J]. IEEE Transactions on Artificial Intelligence, 2021, 2(6): 528-536.
- [12] QING C J, CAI B, YANG Q Y, et al. Deep learning for CSI feedback based on superimposed coding[J]. IEEE Access, 2019, 7: 93723-93733.
- [13] LIU Z Y, ROSARIO M D, DING Z. A Markovian model-driven deep learning framework for massive MIMO CSI feedback[J]. IEEE Transactions on Wireless Communications, 2021, 21(2): 1214-1228.
- [14] GUO J H, WANG L, LI F, et al. CSI feedback with model-driven deep learning of massive MIMO systems[J]. IEEE Communications Letters, 2021, 26(3): 547-551.
- [15] DONOHO D L, MALEKI A, MONTANARI A. Message-passing algorithms for compressed sensing[J]. Proceedings of the National Academy of Sciences, 2009, 106(45): 18914-18919.
- [16] BALATSOUKAS-STIMMING A, STUDER C. Deep unfolding for communications systems: a survey and some new direc-

- tions[C]//Proceedings of IEEE International Workshop on Signal Processing Systems (SIPS). Piscataway: IEEE Press, 2019: 266-271.
- [17] MONTANARI A, ELDAR Y C, KUTYNIOK G. Graphical models concepts in compressed sensing[M]. London: Cambridge University, 2012.
- [18] LOHIT S, KULKARNI K, KERVICHE R, et al. Convolutional neural networks for noniterative re-construction of compressively sensed images[J]. IEEE Transactions on Computational Imaging, 2018, 4(3): 326-340.
- [19] FRANOIS C. Xception: deep learning with depthwise separable convolutions[C]//Proceedings of The IEEE International Conference on Computer Vision and Pattern Recognition: Piscataway: IEEE Press, 2017: 1251-1258.
- [20] ZHANG K, ZUO W M, CHEN Y J, et al. Beyond a gaussian denoiser: residual learning of deep CNN for image denoising[J]. IEEE Transactions on Image Processing, 2017, 26(7): 3142-3155.
- [21] LIU L F, OESTGES C, POUTANEN J, et al. The COST 2100 MIMO channel model[J]. IEEE Wireless Communications, 2012, 19(6): 92-99.
- [22] ZHANG Z J. Improved Adam optimizer for deep neural networks[C]//Proceedings of IEEE/ACM 26th International Symposium on Quality of Service. Piscataway: IEEE Press, 2018: 1-2.

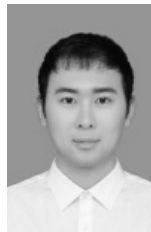
[作者简介]



廖勇 (1982-), 男, 四川自贡人, 博士, 重庆大学副研究员, 主要研究方向为下一代无线通信、人工智能及其在无线通信中的应用等。



程港 (1997-), 男, 重庆万州人, 重庆大学硕士生, 主要研究方向为智能信号与信息处理。



李玉杰 (1996-), 男, 河南周口人, 重庆大学硕士生, 主要研究方向为智能信号与信息处理。