模型驱动深度学习用于非相干大规模机器类通信

Zhe Ma、Wen Wu，*IEEE*高级会员，Feifei Gao，*IEEE*会士，以及 Xuemin (Sherman) Shen，*IEEE*会士

arXiv:2301.00516v1 [cs.IT] 2 Jan 2023

摘要

本文研究了采用单阶段非相干方案的大规模机器类通信（mMTC）中的设备活动与数据联合检测问题。在该方案中，数据比特嵌入导频序列，基站无需进行显式信道估计，即可同时检测活跃设备及其嵌入的数据比特。由于非相干传输方案引入了相关稀疏模式，传统的近似消息传递（AMP）算法无法实现令人满意的性能。因此，我们提出了一种深度学习（DL）改进型AMP网络（DL-mAMPnet），通过有效利用导频活动相关性来提升检测性能。DL-mAMPnet通过将AMP算法展开为前馈神经网络构建而成，将AMP算法的原理性数学模型与强大的学习能力相结合，从而兼具两种技术的优势。在DL-mAMPnet中引入了可训练参数，用于近似相关稀疏模式和大尺度衰落系数。此外，我们设计了一个细化模块，利用相关稀疏模式所产生的空间特征，进一步提升性能。仿真结果表明，在符号错误率性能方面，所提出的方法DL-mAMPnet能够显著优于传统算法。

索引词

大规模机器类通信（mMTC）、非相干传输、免授权随机接入、深度学习、模型驱动。

Z. Ma 和 F. Gao 任职于清华大学人工智能研究院、智能技术与系统国家重点实验室、北京信息科学与技术国家研究中心、清华大学自动化系，中国北京 100084（电子邮箱：maz16@mails.tsinghua.edu.cn；feifeigao@ieee.org）。

W. Wu 任职于鹏城实验室前沿研究中心，中国广东省深圳市 518055（电子邮箱：wuw02@pcl.ac.cn）。

X. Shen 任职于加拿大安大略省滑铁卢市 N2L 3G1 滑铁卢大学电气与计算机工程系（电子邮箱：sshen@uwaterloo.ca）。

I. 引言

为迎接物联网（IoT）的时代，第三代合作伙伴计划（3GPP）已将海量机器类通信（mMTC）列为第五代（5G）及后续网络的三大主要服务类别之一 [1]。在典型的大规模机器类通信场景中，大量物联网设备需要与单个基站（BS）建立以上行为主的通信 [2]。上行传输通常具有突发性，且数据包大小较短，因此只有少量设备会随机地短暂活跃 [3]-[4]。因此，传统的基于授权的随机接入协议不适用于大规模机器类通信场景。为更好地支持大规模机器类通信服务，一种可行的方案是开发新型多址方案，以便及时、准确地完成用户活跃检测与数据检测。

免授权（GF）随机接入是大规模机器类通信和物联网的一种前景广阔的解决方案，因为它消除了基站与海量设备之间协调所需的信令开销 [5]。在免授权随机接入中，用户活跃检测与数据检测通常通过两阶段相干方案进行。具体而言，每个被激活的设备直接发送唯一的导频序列，随后发送数据包，而无需事先进行调度分配。基站收到这些设备的叠加信号后，首先检测活跃设备并估计信道，然后据此解码相应的传输数据比特。然而，由于设备数量庞大，不可能为每个设备分配正交导频序列，这不可避免地会导致设备间发生碰撞并造成性能下降 [6]。得益于大规模机器类通信流量的稀疏特性，设备活动检测和信道估计可以建模为压缩感知（CS）问题 [7]。因此，研究人员已将各种压缩感知技术用于大规模机器类通信中的设备检测，且结果表明，这些技术能够通过缓解导频污染，优于传统方法 [8]-[10]。然而，两阶段相干方案会产生不可忽略的信道训练开销。因此，该方案可能不适用于设备通常间歇性发送小数据包的大规模机器类通信场景，这促使研究人员考虑非相干方案 [11]-[16]。

已有多项研究尝试探讨单阶段非相干方案 [13]-[16]。与相干方案不同，非相干方案无需进行显式信道估计。单阶段非相干方案的基本思路是为每个设备分配多个不同的导频序列。传输时，每个设备根据其数据仅选择

一个导频序列，而基站通过判定接收到的是哪个导频序列，联合检测用户活动与数据。论文 [13] 提出了一种在导频序列中嵌入 1 比特的新方法，其性能优于两阶段相干方案。文献 [14] 考虑了嵌入多个比特的情况，并使用近似消息传递（AMP）算法联合进行用户活跃检测与数据检测。文献 [15], 提出了一种改进的 AMP 算法，利用软阈值函数从可能的导频序列中选出一个，同时抑制其他序列。文献 [16],提出了一种基于协方差的检测方案，用于获取所传输导频序列的索引。然而，以上所有研究均假设每个导频序列的活动相互独立且同分布。尽管独立同分布假设能够得到解析上易于处理的解，但它忽略了同一用户各导频序列活动之间的相关性，因此可能不是最优的。本文研究利用深度学习方法探索稀疏模式的相关结构，并改进用户活跃检测与数据检测的可行性。

凭借解决复杂且棘手问题的强大能力，机器学习已成为未来无线通信领域备受青睐的研究方向 [17]- [24]。尤其是，作为机器学习的一个重要分支，深度学习已被广泛用于信号检测 [20], 、信道估计 [21], 和星座设计 [22] ，以提升性能并降低计算复杂度。在无线通信领域众多采用深度学习的技术中，将迭代算法展开为深度神经网络（DNN）的“深度展开”方法尤为引人关注 [23]。通过将通信领域的专家知识融入深度神经网络，“深度展开”继承了经典算法的数学模型，并使网络拓扑设计具有可解释性 [24]。与此同时，借助深度学习（DL）强大的学习能力，“深度展开”能够弥补模型不准确及预设参数所造成的缺陷。

受现有研究启发，我们提出一种模型驱动的深度学习算法，即深度学习改进的AMP网络（DL-mAMPnet），用于单阶段非相干方案下的大规模机器类通信中的设备活动与数据联合检测。DL-mAMPnet通过展开近似消息传递算法构建，并加入可训练参数和细化模块，以利用导频序列活动的相关稀疏模式。仿真结果验证了所提出的DL-mAMPnet在符号错误率（SER）方面的优越性能。主要贡献总结如下。

- 我们将单阶段非相干方案下大规模机器类通信中的设备活动与数据联合检测建模为具有两级稀疏性的分层压缩感知问题，其中，设备活动稀疏性和发送导频序列稀疏性分别建模为系统级稀疏性和设备级稀疏性。

- 我们提出一种基于 AMP 的算法来解决所构建的压缩感知问题。在此基础上，我们讨论该算法的局限性，这也构成了设计基于深度学习的算法的根本动机。

- 我们提出一种称为 DL-mAMPnet 的基于深度学习的算法，用于联合进行设备活动检测和数据检测。DL-mAMPnet 由多个 AMP 层和一个细化模块组成。AMP 层是将 AMP 算法展开为前馈深度神经网络而得到的，其中引入了可训练参数，以补偿传统 AMP 算法中独立同分布模型的不准确性。细化模块利用两级稀疏结构独特的空间特征，对 AMP 层的输出进行细化。

本文其余部分安排如下：第二节介绍系统模型，并简要说明非相干方案。第三节构建具有两级稀疏性的分层压缩感知问题，并据此推导一种基于 AMP 的算法。第四节详细阐述所提出的 DL-mAMPnet 的结构。第五节介绍所提出的 DL-mAMPnet 的参数初始化和训练方法。第六节给出仿真结果，第七节总结全文。

符号说明：我们分别使用普通小写字母、粗体小写字母和粗体大写字母表示标量、向量和矩阵。对于矩阵 $\boldsymbol{X}$，$\boldsymbol{X}^T$ 表示其转置，$\boldsymbol{X}^H$ 表示其共轭转置，$|\boldsymbol{X}|$ 表示其行列式，$||\boldsymbol{X}||_F$ 表示其弗罗贝尼乌斯范数。对于向量 $\boldsymbol{x}$，$||\boldsymbol{x}||_p$ 表示其 l_p范数。$\mathbb{E}\{\cdot\}$ 表示期望运算。$\mathbb{R}^{M\times N}$ 和 $\mathbb{C}^{M\times N}$分别表示 $M\times N$维实空间和复空间。$\mathcal{CN}(\boldsymbol{\mu},\ \boldsymbol{\Sigma})$表示均值为 $\boldsymbol{\mu}$ 、协方差为 $\boldsymbol{\Sigma}$的多元复高斯分布。

II. 系统模型

A. 大规模机器类通信系统中的上行海量接入场景

我们考虑大规模机器类通信系统中的典型上行海量接入场景，其中一组随机分布的单天线设备（记为 $\mathcal{N}=\{1,\cdots,N\}$）与

一台配备 M 根天线的基站通信。从设备 n 到基站的上行信道记为 $\boldsymbol{h}_n \in \mathbb{C}^{M\times 1}$，其模型为

$$\boldsymbol{h}_n = \sqrt{\beta_n}\boldsymbol{g}_n, \forall n \in \mathcal{N}, \tag{1}$$

其中，β_n 是大尺度衰落分量，$\boldsymbol{g}_n$ 表示小尺度衰落分量。我们假设 $\boldsymbol{g}_n$服从 $\mathcal{CN}$（$\mathbf{0}$，$\boldsymbol{I}_M$）分布，因此有 $\boldsymbol{h}_n \sim \mathcal{CN}$（$\mathbf{0}$，$\beta_n\boldsymbol{I}_M$）。本文采用块衰落信道模型，其中 $\boldsymbol{h}_n$ 在信道相干时间内保持不变，但在不同块之间相互独立。

由于大规模机器类通信的稀疏活动模式，每个块中只有少量设备处于活跃状态。我们假设设备已实现同步，并且每个设备在每个块中以概率 ϵ 独立决定是否接入信道。因此，设备 $n \in \mathcal{N}$ 的设备活动指示变量定义为

$$\alpha_n = \begin{cases} 1, & \text{if device } n \text{ is active,} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

其中，$\Pr(\alpha_n = 1) = \epsilon$，且$\Pr(\alpha_n = 0) = 1 - \epsilon$。我们进一步将一个块内的活跃设备集合定义为

$$\mathcal{K} = \{n \in \mathcal{N} : \alpha_n = 1\}, \tag{3}$$

活跃设备的数量为 $K = |\mathcal{K}|$。基站处的接收信号 $\boldsymbol{y} \in \mathbb{C}^{M\times 1}$ 表示为

$$\boldsymbol{y} = \sum_{n\in\mathcal{N}} \alpha_n \boldsymbol{h}_n x_n + \boldsymbol{n} = \sum_{k\in\mathcal{K}} \boldsymbol{h}_k x_k + \boldsymbol{n}, \tag{4}$$

其中，$x_n \in \mathbb{C}$ 是设备 n发送的信号，$\boldsymbol{n} \in \mathbb{C}^{M\times 1}$ 是服从 $\mathcal{CN}$ 分布的加性高斯白噪声（AWGN）（$\mathbf{0}, \sigma^2\boldsymbol{I}_M$）。

B. 单阶段非相干方案

为了成功传输活跃设备的消息，文献中提出了两种方案，即两阶段相干方案和单阶段非相干方案。两阶段相干方案将每个相干块划分为两个连续的阶段。在第一阶段，活跃设备同步向基站发送导频序列，基站联合检测设备活动（即 α_n）以及相应的信道（即 $\boldsymbol{h}_n$、$\forall n \in \mathcal{K}$）。在第二阶段，活跃设备利用

剩余的相干块向基站发送消息，基站则根据第一阶段获得的设备活动和信道信息对这些消息进行解码。

与两阶段相干方案不同，本文考虑的单阶段非相干方案无需显式信道估计，即可联合检测活跃设备及其对应的消息。具体而言，在非相干方案中，传输消息嵌入在每个活跃设备所传输的导频序列索引中。为此，每个设备都维护一组唯一的预先分配的 $Q = 2^J$ 导频序列。当设备处于活跃状态时，它通过发送该序列集合中的一个序列来传输 J比特消息。通过检测接收到的是哪些序列，基站可以获取活跃设备的身份，以及每个活跃设备发送的 J比特消息。我们将分配给设备 n 的导频序列定义如下:

$$\boldsymbol{S}_n = \{\boldsymbol{s}_n^1, \boldsymbol{s}_n^2, \cdots, \boldsymbol{s}_n^Q\}, \tag{5}$$

其中 $\boldsymbol{s}_n^q = [s_{n_1}^q, s_{n_2}^q, \cdots, s_{n_L}^q]^T \in \mathbb{C}^{L\times 1}$， $1 \le q \le Q$，且 L 为序列长度。注意，导频序列总数通常远大于导频序列的长度（或相干块的长度），即 $NQ \gg L$。因此，不可能为所有设备分配两两正交序列。遵循开创性工作 [25], ，本文采用随机高斯序列。具体而言，导频序列的每个元素均由零均值、方差为 $1/L$的独立同分布复高斯分布生成，即 $s_{n_l}^q \sim \mathcal{CN}(0,\ 1/L)$，从而使每个导频序列具有单位范数，即 $||\boldsymbol{s}_n^q||_2 = 1,\ \ \forall n \in \mathcal{N}$且 $q = 1, \cdots,\ Q$。

传输时，每个活跃设备根据其消息，从 $\boldsymbol{S}_n$ 中恰好选择一个序列。于是，非相干方案的合成接收信号 $\boldsymbol{Y} \in \mathbb{C}^{L\times M}$ 可表示为

$$\boldsymbol{Y} = \sum_{n=1}^{N}\sum_{q=1}^{Q} \alpha_n^q \boldsymbol{s}_n^q \boldsymbol{h}_n^T + \boldsymbol{N} = \sum_{n=1}^{N} \boldsymbol{S}_n \boldsymbol{X}_n + \boldsymbol{N}, \tag{6}$$

其中，$\boldsymbol{X}_n = [\alpha_n^1 \boldsymbol{h}_n, \alpha_n^2 \boldsymbol{h}_n, \cdots, \alpha_n^Q \boldsymbol{h}_n]^T \in \mathbb{C}^{Q\times M}$ 和 $\alpha_n^q \in \{0, 1\}$ 表示设备 n 是否发送序列 q，这里稍微滥用了符号。回顾一下，每个设备以概率 ϵ 激活，我们有

$$\sum_{q=1}^{Q} \alpha_n^q = \begin{cases} 1, & \text{with probability } \epsilon; \\ 0, & \text{with probability } 1-\epsilon. \end{cases} \tag{7}$$

进一步将所有 N设备的序列拼接为 $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_N] \in \mathbb{C}^{L\times NQ}$后，式 (6) 中的接收信号可简化为

$$\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{X} + \boldsymbol{N}, \tag{8}$$

其中$\boldsymbol{X}=[\boldsymbol{X}_1^T,\boldsymbol{X}_2^T,\cdots,\boldsymbol{X}_N^T]^T\in\mathbb{C}^{NQ\times M}$。式 (8) 的示意图如图 1 所示，直观地展示了 $\boldsymbol{X}$ 具有层次稀疏结构。层次稀疏结构包含两个层次的稀疏性，即系统级稀疏性 和设备级稀疏性。系统级稀疏性意味着 $\boldsymbol{X}$ 中的大多数行均为零，这是由突发流量模式造成的。设备级稀疏性则要求 $\boldsymbol{X}_n$，$\forall n$中至多存在一行非零，这是因为每个活跃设备只会从其导频集合中发送一个导频序列。

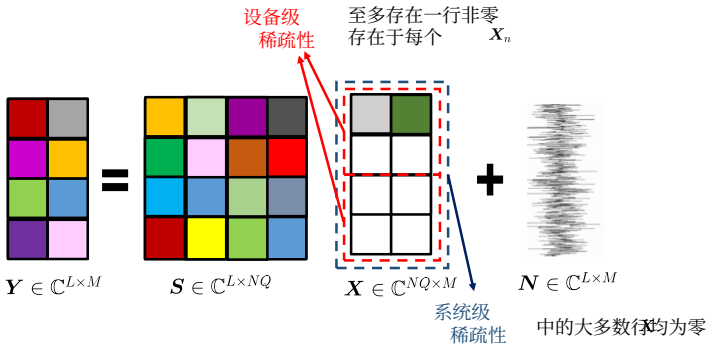


图1. 信号模型示意图。

III. 问题表述与基于AMP的联合检测算法

A. 问题表述

我们的目标是检测二元变量 α_n^q，它同时表示设备 n 是否活跃以及其发送的消息。为此，可以从接收信号 $\boldsymbol{Y}$中恢复 $\boldsymbol{X}$。恢复出 $\boldsymbol{X}$ 后，即可根据 $\boldsymbol{X}$的各行确定 α_n^q。由于 $\boldsymbol{X}$ 具有层次稀疏结构，这类问题是一个已知测量矩阵 $\boldsymbol{S}$ 的经典压缩感知问题。因此，我们可以将该问题表述如下：

$$\mathcal{P}1:\min_{\boldsymbol{X}}\quad||\boldsymbol{Y}-\boldsymbol{S}\boldsymbol{X}||_F^2 \tag{9}$$

$$s.t.\quad\sum_{n=1}^{N}\sum_{q=1}^{Q}\mathbb{I}(\boldsymbol{X}_{nq,:})\leq K, \tag{10}$$

$$\sum_{q=1}^{Q}\mathbb{I}(\boldsymbol{X}_{nq,:})\leq 1,\forall n, \tag{11}$$

其中 $\boldsymbol{X}_{nq,:}$ 是 $\boldsymbol{X}_n$ 的第 q行，而 $\mathbb{I}(\cdot)$ 是定义为 a 的指示函数 s

$$\mathbb{I}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \text{ has non-zero elements;} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

约束（10）源于系统级稀疏性，而约束（11）确保设备级稀疏性。然而，由于约束不可微，直接求解 $\mathcal{P}1$ 颇具挑战。因此，我们将指示函数替换为 l_2 范数，将（11）放宽为$l_{2,1}$-范数正则化最小二乘问题，如下所示 [26]

$$\min_{\boldsymbol{X}} \quad \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{S}\boldsymbol{X}||_F^2 + \lambda \sum_{n=1}^{N}\sum_{q=1}^{Q} ||\boldsymbol{X}_{nq,:}||_2, \tag{13}$$

其中， λ 是可调参数，用于平衡解的稀疏性与均方误差（MSE） $||\boldsymbol{Y} - \boldsymbol{S}\boldsymbol{X}||_F^2$。尽管正交匹配追踪（OMP）和稀疏贝叶斯学习（SBL）等传统压缩感知算法可直接用于求解（13），但它们因矩阵求逆操作而具有较高的计算复杂度，在设备数量庞大的大规模机器类通信系统中尤为如此。鉴于此，本文采用计算高效的AMP算法作为主要技术 [27]。

*B. AMP*算法综述

AMP是用于高维问题中统计估计的一类高效算法，例如线性回归和低秩矩阵估计。AMP算法的目标是基于 $\boldsymbol{Y}$，获得 $\boldsymbol{X}$ 的均方误差最小的估计值。从$\boldsymbol{X}_0 = \boldsymbol{0}$ 和 $\boldsymbol{R}_0 = \boldsymbol{Y}$ 出发，AMP算法可描述如下:

$$\boldsymbol{X}_{t+1,n} = \eta_{t,n}(\boldsymbol{S}_n^H \boldsymbol{R}_t + \boldsymbol{X}_{t,n}), \forall n, \tag{14}$$

$$\boldsymbol{R}_{t+1} = \boldsymbol{Y} - \boldsymbol{S}\boldsymbol{X}_{t+1} + b_t \boldsymbol{R}_t, \tag{15}$$

其中 $t = 0$，1，$\cdots$是迭代的索引，$\eta_{t,n}(\cdot)$ 是设备 n的收缩函数，可将其输入中的某些元素收缩为零，而 $\boldsymbol{R}_t$ 是相应的残差。(15) 式中的残差通过“Onsager 校正”项 $b_t\boldsymbol{R}_t$进行更新，这显著提升了 AMP算法 [28]的性能。注意，$\eta_{t,n}(\cdot)$ 假设为 Lipschitz 连续，且 b_t 可写为

$$b_t = \frac{1}{L}\sum_{n=1}^{N} \eta_{t,n}^{'}(\boldsymbol{S}_n^H \boldsymbol{R}_t + \boldsymbol{X}_{t,n}), \tag{16}$$

其中 $\eta_{t,n}'(\cdot)$ 是 $\eta_{t,n}(\cdot)$ 的一阶导数。除了提升性能之外，Onsager 校正还使得 AMP算法能够通过渐近区间 [29]中的一组状态演化方程进行分析。当 L，$N \to \infty$，且二者之比收敛于一个正数，即 $N/L \to \rho$ 其中 $\rho \in (0, \infty)$，同时保持数据长度 J 不变时，便处于渐近区间。为便于理论分析，本文考虑一种特定的渐近区间，其中 $N \to \infty$，且大尺度衰落分量 β_n' 的经验分布收敛于一个固定分布 p_β。

定义 $\beta \sim p_\beta$和 $\boldsymbol{X}_\beta \in \mathbb{C}^{Q\times M}$为服从 $(1-\frac{\epsilon}{Q})\prod_{i=1}^{Q}\delta_{\boldsymbol{x}_{\beta,i}}+\frac{\epsilon}{Q}\sum_{i=1}^{Q}P_{\boldsymbol{h}_\beta}\prod_{j\neq i}\delta_{\boldsymbol{x}_{\beta,j}}$分布的随机矩阵，其中 $\delta_{\boldsymbol{x}_{\beta,i}}$是对应于元素 $\boldsymbol{x}_{\beta,i}$的零点处的狄拉克δ函数，而 $P_{\boldsymbol{h}_\beta}$表示分布 $\boldsymbol{h}_\beta \sim \mathcal{CN}(\boldsymbol{0}, \beta\boldsymbol{I}_M)$。状态演化方程可写为关于 $t \geq 0$ [29]的以下递推式

$$\boldsymbol{\Sigma}_0 = \sigma^2\boldsymbol{I}_M + \rho\mathbb{E}_\beta\{\boldsymbol{X}_\beta^H\boldsymbol{X}_\beta\}, \tag{17}$$

$$\boldsymbol{\Sigma}_{t+1} = \sigma^2\boldsymbol{I}_M + \rho\mathbb{E}_\beta\{(\eta_t(\boldsymbol{X}_\beta + \boldsymbol{V}\boldsymbol{\Sigma}_t^{\frac{1}{2}}) - \boldsymbol{X}_\beta)^H(\eta_t(\boldsymbol{X}_\beta + \boldsymbol{V}\boldsymbol{\Sigma}_t^{\frac{1}{2}}) - \boldsymbol{X}_\beta)\}, \tag{18}$$

其中 $\boldsymbol{V} \in \mathbb{C}^{Q\times M}$ 是一个与 $\boldsymbol{X}_\beta$相互独立的随机矩阵，其各行独立同分布，且每一行均服从分布 $\mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_M)$。由式（14）和（18）可知，对 $\boldsymbol{S}_n^H\boldsymbol{R}_t + \boldsymbol{X}_{t,n}$ 应用$_1\eta_{t,n}(\cdot)$，在统计上等价于对 $\boldsymbol{X}_{t,n} + \boldsymbol{V}\boldsymbol{\Sigma}_t^2$应用 $\eta_{t,n}(\cdot)$。因此，收缩函数$\eta_{t,n}(\cdot)$的输入可建模为加性高斯白噪声污染信号，即,

$$\boldsymbol{Z}_{t,n} = \boldsymbol{X}_{t,n} + \boldsymbol{S}_n^H\boldsymbol{R}_t = \boldsymbol{X}_{t,n} + \boldsymbol{V}\boldsymbol{\Sigma}_t^{\frac{1}{2}}, \tag{19}$$

在这种情况下，式 (14) 给出的更新在统计上等价于一个去噪问题，因此 $\eta_t(\cdot)$ 也可以称为“去噪器”。下文为方便起见，将“收缩函数”和“去噪器”互换使用。

C. 基于近似消息传递的设备活动与日期联合检测算法

联合检测算法的核心思想是首先根据 $\boldsymbol{Y}$估计出 $\boldsymbol{X}$ ，在此基础上，再根据 $\boldsymbol{X}$中各行的范数确定 α_n^q 。为此，我们首先在最小均方误差最优准则下推导去噪器 $\eta_{t,n}(\cdot)$。随后，我们观察到$\eta_{t,n}(\cdot)$呈现出渐近特性，这促使我们设计一种基于阈值的策略，从 $\boldsymbol{X}$中提取 α_n^q。

1) 的推导$\eta_n(\cdot)$：为简化记号，下面省略迭代索引 t 。根据（19），给定 $\boldsymbol{X}_n$ 时 $\boldsymbol{Z}_n$ 的似然可表示为

$$P_{\boldsymbol{Z}_n|\boldsymbol{X}_n} = \prod_{q=1}^{Q} \frac{\exp(-(\boldsymbol{z}_n^q - \boldsymbol{x}_n^q)^H \boldsymbol{\Sigma}^{-1} (\boldsymbol{z}_n^q - \boldsymbol{x}_n^q))}{\pi^M |\boldsymbol{\Sigma}|}. \tag{20}$$

因此，MMSE最优去噪器由条件期望 $\mathbb{E}\{\boldsymbol{X}_n|\boldsymbol{Z}_n\}$给出，可表示为

$$\eta_n(\boldsymbol{Z}_n) = \mathbb{E}\{\boldsymbol{X}_n|\boldsymbol{Z}_n\} = [\phi_n^1 \boldsymbol{\Omega}_n \boldsymbol{z}_n^1, \cdots, \phi_n^Q \boldsymbol{\Omega}_n \boldsymbol{z}_n^Q], \tag{21}$$

其中

$$\boldsymbol{\Omega}_n = \beta_n (\beta_n \boldsymbol{I}_M + \boldsymbol{\Sigma})^{-1}, \tag{22}$$

$$\phi_n^q = \frac{1}{1 + \frac{Q-\epsilon}{\epsilon} \exp(M(\psi_n - \pi_n^q))}, \tag{23}$$

$$\psi_n = \frac{\log(|\boldsymbol{I}_M + \beta_n \boldsymbol{\Sigma}^{-1}|)}{M}, \tag{24}$$

and

$$\pi_n^q = \frac{\boldsymbol{z}_n^{qH} (\boldsymbol{\Sigma}^{-1} - (\boldsymbol{\Sigma} + \beta_n \boldsymbol{I}_M)^{-1}) \boldsymbol{z}_n^q}{M}. \tag{25}$$

证明：请参见附录A。

需要注意的是，MMSE最优去噪器 $\eta_n(\cdot)$ 相当复杂，因为它涉及状态演化矩阵 $\boldsymbol{\Sigma}$ 的计算，其中需要进行矩阵乘法和求期望。因此，我们利用以下定理简化 $\eta_n(\cdot)$。

定理 *1*：考虑渐近区间：设备数量 N 和导频序列长度 L 均趋于无穷大，且二者之比收敛于某个固定正值，即 $N/L \to \rho$，其中 $\rho \in (0, \infty)$。经过每次迭代后，状态演化矩阵 $\boldsymbol{\Sigma}_t$ 始终保持为对角线元素相同的对角矩阵，即

$$\boldsymbol{\Sigma}_t = \tau_t^2 \boldsymbol{I}_M, \ \forall t \geq 0. \tag{26}$$

相应地，式 (19) 给出的信号模型可简化为

$$\boldsymbol{Z}_{t,n} = \boldsymbol{X}_{t,n} + \boldsymbol{S}_n^H \boldsymbol{R}_t = \boldsymbol{X}_{t,n} + \tau_t \boldsymbol{V}, \tag{27}$$

且式 (21)–(25) 中给出的均方误差最优去噪器可简化为

$$\eta_n(\boldsymbol{Z}_n) = \mathbb{E}\{\boldsymbol{X}_n|\boldsymbol{Z}_n\} = [\phi_n^1 \omega_n \boldsymbol{z}_n^1, \cdots, \phi_n^Q \omega_n \boldsymbol{z}_n^Q], \tag{28}$$

其中

$$\omega_n = \frac{\beta_n}{\beta_n + \tau^2}, \tag{29}$$

$$\phi_n^q = \frac{1}{1 + \frac{Q-\epsilon}{\epsilon}\exp(M(\psi_n - \pi_n^q))}, \tag{30}$$

$$\psi_n = \log(1 + \frac{\beta_n}{\tau^2}), \tag{31}$$

and

$$\pi_n^q = \frac{\beta_n \boldsymbol{z}_n^{qH} \boldsymbol{z}_n^q}{\tau^2(\beta_n + \tau^2)M}. \tag{32}$$

最后， τ_t^2可以通过以下关于 $t \geq 0$的递推式得到:

$$\tau_0^2 = \sigma^2 + \rho\epsilon\mathbb{E}_\beta\{\beta\}, \tag{33}$$

$$\tau_{t+1}^2 = \sigma^2 + \rho\sum_{q=1}^{Q}\mathbb{E}_\beta\{\frac{\phi_\beta^q \beta\tau_t^2}{\beta + \tau_t^2}\} + \rho\sum_{q=1}^{Q}\mathbb{E}_\beta\{\phi_\beta^q(1-\phi_\beta^q)\frac{\beta^2 \boldsymbol{z}_n^{qH}\boldsymbol{z}_n^q}{(\beta+\tau_t^2)^2 M}\}. \tag{34}$$

为简洁起见，此处省略详细证明。感兴趣的读者可参阅 [8],中的定理 1，其中给出了类似的推导。需要指出的是，本文所提出的定理 1 本质上是对 [8]中定理 1的推广。当每个设备仅分配一个导频序列时，即 $Q = 1$，所提出的定理 1可化为 [8]中的定理 1。

2）基于阈值的策略：由式（28）—（30）可知，当 M较大时，若 $\pi_n^q > \psi_n$ ，则有 $\phi_n^q \to 1$；若 $\pi_n^q < \psi_n$，则有 $\phi_n^q \to 0$ 。 ϕ_n^q 的渐近行为表明，采用基于阈值的策略进行解的细化是合理的。同时，考虑到式（7）中的设备稀疏性，需要执行元素选择操作，将每个 $\boldsymbol{X}_n$ 中除幅度最大元素之外的所有元素置零。因此，所提出的基于阈值的策略应能够执行以下两项操作。

元素选择操作：为确保满足式（11）中的稀疏性约束，我们在每个 $\boldsymbol{X}_n = [\boldsymbol{x}_n^1, \boldsymbol{x}_n^2, \cdots, \boldsymbol{x}_n^Q]$ 中选择最大的一行，并将最大元素的索引定义为

$$i_n^* = \arg\max_i \boldsymbol{x}_n^{i\,H}\boldsymbol{x}_n^i, \forall n \in \mathcal{N}. \tag{35}$$

基于阈值的判定操作：获得 i_n^* 后，二元变量向量 $\boldsymbol{\alpha}_n = \{\alpha_n^1, \cdots, \alpha_n^Q\}$ 可表示为

$$\boldsymbol{\alpha}_n = \begin{cases} \boldsymbol{e}_{i_n^*}, & \text{if } \kappa_n^{i_n^*} > 0; \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{36}$$

其中，$\boldsymbol{e}_{i_n^*}$ 是一个长度为 Q 的独热向量，只有第 i_n^* 个元素等于 1，其余元素均等于 0；相应的阈值根据 (31) 和 (32) 计算为

$$\kappa_n^{i_n^*} = \frac{\boldsymbol{z}_n^{i_n^*H}\boldsymbol{z}_n^{i_n^*}\beta_n}{\tau_t^2(\beta_n+\tau_t^2)M} - \log\left(1+\frac{\beta_n}{\tau_t^2}\right). \tag{37}$$

3）局限性：尽管传统的基于AMP的算法能够成功地从 $\boldsymbol{Y}$中恢复 a_n^q，但它存在一些固有的局限性：（i）传统 AMP 算法隐含地假设$\boldsymbol{X}_n$ 的先验分布具有独立同分布的元素，忽略了设备级稀疏性所导致的 $\boldsymbol{X_n}$ 各行之间的相关性；（ii）计算去噪器$\eta_{t,\ n}(\cdot)$和阈值 κ_n 需要知道 β_n的精确值，而在拥有海量设备的大规模大规模机器类通信系统中，获取该值的成本很高。

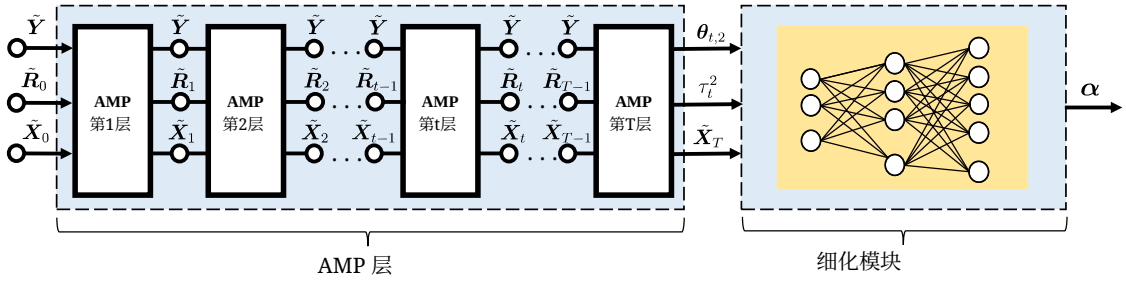


图2. 所提出的DL-mAMPnet网络架构。

IV. 深度学习改进的AMP网络

为了解决上述局限性，我们提出了一种深度学习改进的 AMP 网络（DL-mAMPnet）。DL-mAMPnet 通过将 AMP 算法展开为前馈深度神经网络构建而成，继承了 AMP 算法的数学模型和结构，因此无需精确建模。在此基础上，我们在 DL-mAMPnet 中引入少量可训练参数，以学习激活概率和大尺度衰落。通过将激活概率设为可训练参数，我们补偿了传统 AMP 算法中独立同分布假设所导致的不准确性。通过将大尺度衰落系数设为可训练参数，我们无需对大规模设备的大尺度衰落进行统计测量。根据第三节-C 中的基于阈值的策略，我们进一步设计了一个细化模块，以保证设备级稀疏性并获得所需的a_n^q。

如图2所示，所提出的 DL-mAMPnet 由 T 个统一 AMP 层和一个细化模块组成。为清晰起见，以下小节将分别详细阐述 DL-mAMPnet 的各个部分。

A. 输入与输出

为便于 DL-mAMPnet 的学习过程，需要将复矩阵转换到实数域，然后进行向量化。为此，我们首先将 (8) 表示为

$$\begin{bmatrix} \Re(\boldsymbol{Y}) \\ \Im(\boldsymbol{Y}) \end{bmatrix} = \begin{bmatrix} \Re(\boldsymbol{S}) & -\Im(\boldsymbol{S}) \\ \Im(\boldsymbol{S}) & \Re(\boldsymbol{S}) \end{bmatrix} \begin{bmatrix} \Re(\boldsymbol{X}) \\ \Im(\boldsymbol{X}) \end{bmatrix} + \begin{bmatrix} \Re(\boldsymbol{N}) \\ \Im(\boldsymbol{N}) \end{bmatrix}, \tag{38}$$

其中，$\Re(\cdot)$ 和 $\Im(\cdot)$ 分别表示实部和虚部。随后将实部和虚部拼接在一起，并向量化为

$$\tilde{\boldsymbol{Y}} = \text{vec}([\Re(\boldsymbol{Y})^T, \Im(\boldsymbol{Y})^T]^T) \in \mathbb{R}^{2LM\times 1}, \tag{39}$$

$$\tilde{\boldsymbol{S}} = \left[[\Re(\boldsymbol{S}), -\Im(\boldsymbol{S})]^T, [\Im(\boldsymbol{S}), \Re(\boldsymbol{S})]^T\right]^T \otimes \boldsymbol{I}_M \in \mathbb{R}^{2LM\times 2NQM}, \tag{40}$$

$$\tilde{\boldsymbol{X}} = \text{vec}([\Re(\boldsymbol{X})^T, \Im(\boldsymbol{X})^T]^T) \in \mathbb{R}^{2NQM\times 1}, \tag{41}$$

$$\tilde{\boldsymbol{N}} = \text{vec}([\Re(\boldsymbol{N})^T, \Im(\boldsymbol{N})^T]^T) \in \mathbb{R}^{2LM\times 1}, \tag{42}$$

其中，vec(·) 是向量化操作，按列顺序将矩阵展平为向量；$\otimes$ 是克罗内克积算子。因此，(8) 可改写为

$$\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{S}}\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{N}}. \tag{43}$$

根据式 (14)-(15) 中的递归公式，DL-mAMPnet 的输入选为˜接收信号、估计信号和残差，它们的初始值分别为 $\boldsymbol{X}_0 = \boldsymbol{0}$ ˜ ˜ ˜和 $\boldsymbol{R}_0 = \boldsymbol{Y}$。与此同时，与使用 $\boldsymbol{X}$ [30] 的现有受AMP启发的网络不同，我们采用 $\boldsymbol{\alpha} = [\alpha_1^1, \cdots, \alpha_1^Q, \alpha_2^1, \cdots, \alpha_N^Q]^T \in \{0,1\}^{NQ\times 1}$ 作为 DL-mAMPnet 的输出，因此 DL-mAMPnet 训练完成后即可直接得到 α_n^q。

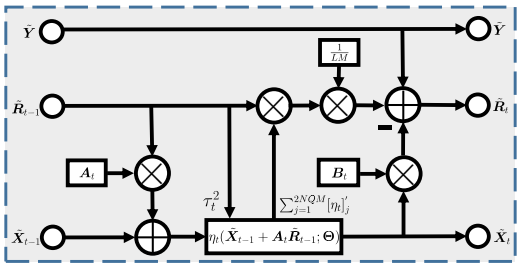


图3. 第 t个近似消息传递层的详细结构。

B. AMP层

由于每个层的结构都相同，我们聚焦于 DL-mAMPnet 的第 t 个 AMP层，˜ ˜其详细结构如图3所示。将输入定义为 $\boldsymbol{X}_{t-1}$、 $\boldsymbol{R}_{t-1}$ ，并将˜ ˜输出定义为 $\boldsymbol{X}_t$、 $\boldsymbol{R}_t$，第 t 个 AMP 层的处理过程如下

$$\tilde{\boldsymbol{X}}_t = \eta_t(\tilde{\boldsymbol{X}}_{t-1} + \boldsymbol{B}_t\tilde{\boldsymbol{R}}_{t-1}; \boldsymbol{\Theta}_t), \tag{44}$$

$$\tilde{\boldsymbol{R}}_t = \tilde{\boldsymbol{Y}} - \boldsymbol{A}_t\tilde{\boldsymbol{X}}_t + \frac{\tilde{\boldsymbol{R}}_{t-1}}{LM}\sum_{j=1}^{2NQM}[\eta_t(\tilde{\boldsymbol{X}}_{t-1} + \boldsymbol{B}_t\tilde{\boldsymbol{R}}_{t-1}; \boldsymbol{\Theta}_t)]'_j, \tag{45}$$

其中， $\boldsymbol{A}_t$ 和 $\boldsymbol{B}_t$ 是充当匹配滤波器的可训练矩阵， $\boldsymbol{\Theta}_t = \{\boldsymbol{\theta}_{t,1}, \boldsymbol{\theta}_{t,2}\}$ 是 η_t (·) 的可训练参数集。

需要指出的是，（28）—（32）中的去噪器无法应用于近似消息传递层，因为（39）—（43）中的复数到实数变换和向量化改变了相应矩阵的维度和分布。按照附录A中的相同推导˜，但将 $\boldsymbol{X}$ 视为实值伯努利高斯变量并改变维度后，（28）中的η_t（·）可表示为

$$\begin{aligned}[\eta_t(\tilde{\boldsymbol{Z}})]_j &= \frac{\beta\tilde{\boldsymbol{Z}}_j}{(\beta+\tau_t^2)\left(1+\frac{Q-\epsilon}{\epsilon}\exp(\log(1+\frac{\beta}{\tau_t^2})^{1/2} - \frac{\tilde{\boldsymbol{Z}}_j^2\beta}{2(\beta+\tau_t^2)\tau_t^2})\right)}, \\ &= \frac{\tilde{\boldsymbol{Z}}_j}{(1+\frac{\tau_t^2}{\beta})\left(1+\sqrt{1+\frac{\beta}{\tau_t^2}}\exp(\log(\frac{Q-\epsilon}{\epsilon}) - \frac{\tilde{\boldsymbol{Z}}_j^2}{2(\tau_t^2+\tau_t^4/\beta)})\right)},\end{aligned} \tag{46}$$

˜其中， $\boldsymbol{Z}_j$ 是 $\boldsymbol{Z}$的第 j个元素。

如第II-D节所述， η_t(·)采用独立同分布假设，无法有效挖掘相关稀疏模式。为解决这一问题，我们将 $\log(\frac{Q-\epsilon}{\epsilon})$ 替换为可训练参数$\boldsymbol{\theta_t}$, $_1 = [\theta_{t,1,1}, \cdots, \theta_{t,1,2NQM}]^T \in \mathbb{R}^{2NQM\times 1}$，从而能够学习并近似 $\tilde{\boldsymbol{X}}$ 各元素之间的相关性

与此同时，为避免依赖大尺度衰落的先验信息，我们引入一个可训练参数 $\boldsymbol{\theta_t}$, $_{\mathbf{2}} = [\theta_{t,2,1}, \cdots, \theta_{t,2,2NQM}]^T \in \mathbb{R}^{2NQM\times 1}$，并在式（46）中用它替代 β 。随后可将可训练的 $\eta_t(\cdot)$定义为

$$[\eta_t(\tilde{\boldsymbol{Z}})]_j = \frac{\tilde{\boldsymbol{Z}}_j}{(1+\frac{\tau_t^2}{\theta_{t,2,j}})\left(1+\sqrt{1+\frac{\theta_{t,2,j}}{\tau_t^2}}\exp(\theta_{t,1,j}-\frac{\tilde{\boldsymbol{Z}}_j^2}{2(\tau_t^2+\tau_t^4/\theta_{t,2,j})})\right)}. \tag{47}$$

$\eta_t(\cdot)$的导数因此可表示为

$$[\eta_t(\tilde{\boldsymbol{Z}})]_j' = \frac{[\eta_t(\tilde{\boldsymbol{Z}})]_j}{\partial\tilde{\boldsymbol{Z}}_j} = \frac{1+\sqrt{1+\frac{\theta_{t,2,j}}{\tau_t^2}}\exp(\theta_{t,1,j}-\frac{\tilde{\boldsymbol{Z}}_j^2}{2(\tau_t^2+\tau_t^4/\theta_{t,2,j})})(1+\frac{\tilde{\boldsymbol{Z}}_j^2}{(\tau_t^2+\tau_t^4/\theta_{t,2,j})})}{(1+\frac{\tau_t^2}{\theta_{t,2,j}})\left(1+\sqrt{1+\frac{\theta_{t,2,j}}{\tau_t^2}}\exp(\theta_{t,1,j}-\frac{\tilde{\boldsymbol{Z}}_j^2}{2(\tau_t^2+\tau_t^4/\theta_{t,2,j})})\right)^2}. \tag{48}$$

注意，为避免计算 τ^2中涉及的期望，本文采用了一个经验结果，其中 τ^2 由 $\tilde{\boldsymbol{Z}}$中受扰噪声的标准差估计，即$\tau_t^2 = ||\tilde{\boldsymbol{R}}_t||_2/\sqrt{2LM}$ [30]。

注释 *1*：值得注意的是，式（28）中推导出的去噪器以分段方式运行，即作用于每个 $\boldsymbol{X}_n$的 Q 行，而AMP层中的 $\eta_t(\cdot)$则对 $\boldsymbol{X}$逐行运行。尽管分段方式相比逐行方式能够更好地利用相关性，但在深度神经网络中实现该方式颇具挑战。这是因为，要实现这种分段方式，我们必须构建 N 子层，或在每个AMP层中施加 N 次迭代。前者将大幅扩大网络规模和可训练参数数量，从而降低DL-mAMPnet的可扩展性并阻碍其训练过程。后者将显著增加DL-mAMPnet的计算复杂度，并削弱“深度展开”的优势。还应注意的是，尽管AMP层借助可训练参数能够探索相关稀疏模式，但式（7）中的设备级稀疏约束并不能得到保证。基于这一考虑，我们在细化模块中提出了一种恰当的方法，利用最大池化-最大反池化操作来确保设备级稀疏性，具体细节如下文小节所述。

C. 细化模块

细化模块应能够在不借助显式信道状态信息（CSI）的情况下，从 $\boldsymbol{X}_T$ 中提取a_n^q ，同时确保设备级稀疏性。为实现这些功能，细化模块集成了两个组件，即软阈值去噪组件和硬阈值判决组件。软阈值去噪˜组件旨在利用层次稀疏结构进一步对 $\boldsymbol{X}_T$ 进行去噪。

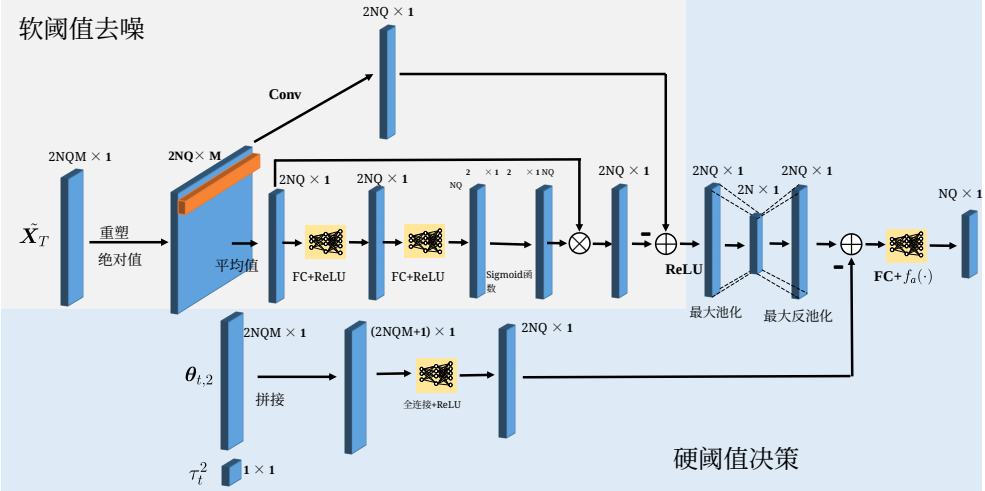


图4. 所提出的细化模块的详细架构。

硬阈值判决组件旨在实现式（35）—（37）中的基于阈值的策略。细化模块的详细结构如图4所示，具体说明如下。

软阈值去噪： 如图1所示，两级稀疏性呈现出一种独特的空间结构，而 AMP 层尚未利用该结构。在此，软阈值˜去噪旨在利用这种空间特征提取 $\boldsymbol{X}_T$ ，在增强有用信息的同时˜去除噪声信息。为此，我们首先对 $\boldsymbol{X}_T$ 进行去向量化，并取绝对值作为

$$\overline{\boldsymbol{X}} = |\text{Vec}^{-1}(\tilde{\boldsymbol{X}}_T)| = [|\Re(\boldsymbol{X})^T|, |\Im(\boldsymbol{X})^T|]^T \in \mathbb{R}^{+2NQ\times M}. \tag{49}$$

然后，对 $\boldsymbol{X}$ 应用卷积核尺寸为 $1 \times M$ 的卷积层，以融合所有 M 根天线的信息，并提取 a_n^q 的粗略估计。此设计基于以下事实：从式 (6) 和图 1 可以看出，$\boldsymbol{X}$ 每一行中的所有 M 个元素共享相同的 a_n^q。粗略估计可表示为 $f_{\theta_c}(\boldsymbol{X})$，其中 $f_{\theta_c}(\cdot)$ 是参数为 θ_c 的卷积层的函数表达式。随后，对 $\boldsymbol{X}$ 应用卷积核尺寸为 $1 \times M$ 的平均池化，以获得跨 M 根天线的 1-D 平均向量。将 1-D 向量 $\boldsymbol{\iota} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{X}_{:,m}$输入双层全连接（FC）网络，以获得缩放参数，从而学习 $\boldsymbol{X}$ 的 $2NQ$ 行中平均值的内部特征。接着，使用 Sigmoid 函数将缩放参数缩放到 (0，1) 范围内，可表示如下

$$\boldsymbol{\vartheta} = \frac{1}{1 + e^{-f_{\theta_{FC_1}}(\boldsymbol{\iota})}}, \tag{50}$$

其中， $\boldsymbol{\vartheta}$ 是缩放向量， $f_{\theta_{FC_1}}(\cdot)$ 是参数为 θ_{FC_1} 的双层全连接网络的函数表达式。接下来，将 $\boldsymbol{\vartheta}$ 与 $\boldsymbol{\iota}$ 相乘，得到阈值：

$$\boldsymbol{\kappa}_{ST} = \boldsymbol{\vartheta} \odot \boldsymbol{\iota}, \tag{51}$$

其中， $\odot$ 是Hadamard积算子。此操作的灵感来自这样一个事实：软阈值处理的阈值必须为正且不能过大 [31]。如果阈值大于 $f_{\theta_c}(\boldsymbol{X})$的最大值，那么软阈值处理的输出将全部为零，从而去除有用信息。最后，将得到的阈值 $\boldsymbol{\kappa}_{ST}$ 减去$f_{\theta_c}(\boldsymbol{X})$，并输入ReLU激活函数，得到

—

$$\boldsymbol{o} = \max(0, f_{\theta_c}(\overline{\boldsymbol{X}}) - \boldsymbol{\kappa}_{ST}), \tag{52}$$

其中， $\boldsymbol{o}$ 表示软阈值去噪组件的输出。从式（52）可以看出，通过将 $\boldsymbol{\kappa}_{ST}$ 保持在合理范围内，可以在消除噪声信息的同时保留有用信息。值得注意的是，这种阈值不是由专家手动设定的，而是可以在所提出的软阈值去噪组件中自动学习，从而无需具备 $\boldsymbol{X}$的信号处理专业知识和统计特性。

—

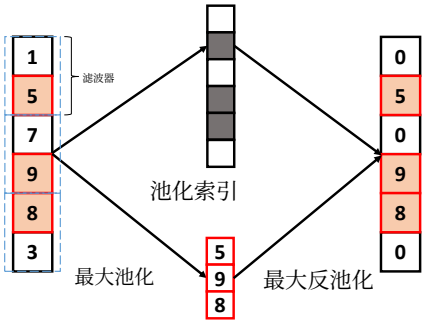


图 5. 最大池化-最大反池化过程示意图。

硬阈值决策： 由于式 (35) 不可微，直接在深度神经网络中实现基于阈值的策略颇具挑战性，而且会阻碍反向传播过程。为解决这一问题，硬阈值判决组件巧妙地利用 最大池化和最大反池化 操作来确保设备级稀疏性。最大池化 是一种下采样技术，它使用最大滤波器对初始输入 [32]中的不重叠子区域进行处理。对于由滤波器表示的每个区域，我们取该区域的最大值，并创建一个新的输出

矩阵，其中每个元素都是原始输入中某个区域的最大值。与之相反，最大反池化通过上采样并用零填充，将最大池化操作的输出扩展至原始大小。除最大值所在位置外，反池化矩阵中的其余元素均补为 0。

为直观地解释这一过程，我们在图5中展示了最大池化 和最大反池化 的过程。从图5可以看出，在每个滤波器中，除最大值保持不变外，其余所有元素均变为0。这种操作完美地实现了式(18)中的元素选择操作。将滤波器大小设为 $Q \times 1$，即可确保 $\boldsymbol{X}_n$的 Q 行中至多只有一行非零，从而保证式(11)中的设备级稀疏约束。还应指出，池化过程只是深度学习过程中用于改变维度大小的一个模块，它没有参数，因此不会影响网络训练。

在保证设备级稀疏性后，接下来需要执行式(36)中的基于阈值的判定操作，即将阈值 $\kappa_n^{j_n^*}$ 与最大池化-最大反池化过程Mp(Mup($\boldsymbol{o}$))得到的矩阵进行比较，以确定二进制序列 $\boldsymbol{\alpha}$ 。然而，确定 $\boldsymbol{\alpha}$时存在一些问题。首先，由于式(37)是在不匹配的独立同分布假设下推导出的，其阈值可能不够精确。为解决这一问题，我们重新审视式(37)，发现阈值是 β 和 τ的函数。由于式(47)已将 β 表示为 $\boldsymbol{\theta}_2$ ，因此我们将$\boldsymbol{\theta}_{T,2}$ 与最后一个AMP层的输出 τ_T^2 拼接起来，并输入一个带有ReLU激活函数的全连接层，以学习准确的阈值，该阈值记为

$$\boldsymbol{\kappa}_{HT} = \max(0, f_{\theta_{FC2}}(\boldsymbol{\theta}_{T,2}, \tau_T^2)), \tag{53}$$

其中， $f_{\theta_{FC2}}$ 是参数为 θ_{FC2}的全连接网络的函数表达式。

然后，将学习得到的阈值 $\boldsymbol{\kappa}_{HT}$ 减去 Mp(Mup($\boldsymbol{o}$))，并输入参数为 θ_{FC3} 的全连接层，以实现基于阈值的判定操作。此处的全连接层具有两项功能：将维度从 $2NQ \times 1$ 压缩至 $NQ \times 1$，以及将 $\boldsymbol{\kappa}_{HT}$-Mp(Mup($\boldsymbol{o}$)) 差值转换为二进制序列。从数学角度来看，用于基于阈值的二元判决的最优函数是符号函数，表示为

$$\text{sng}(x) = \begin{cases} 1, & x > 0; \\ 0, & x \leq 0. \end{cases} \tag{54}$$

然而，由于sng(x)不可微，因此无法用于深度神经网络，这就需要开发一种替代函数。

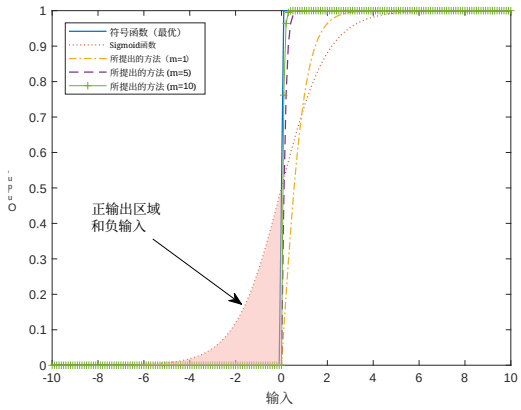


图 6. 最优符号函数、Sigmoid 函数和硬阈值决策函数的曲线。

在基于深度学习的二元决策中，Sigmoid 函数是一种常用选择，在文献中得到了广泛应用 [33], ，因为它可以将输入映射到 [0, 1] 内的区间。然而，Sigmoid 函数仍不适用于硬阈值决策模块。原因如下：(i) Sigmoid 函数的输出是 0 到 1 之间的连续值，这意味着还需要设置一个阈值，才能将输出值区分为 0 或 1。然而，设计合适的阈值通常并非易事；(ii) 根据 (36)，对于负输入，基于阈值的决策输出应严格为 0。然而，如图 6 所示，Sigmoid 函数存在一个区域，即使输入为负，输出仍为正，这可能会引入额外误差。为解决上述问题，我们设计了一种新颖的硬阈值判决函数，其核心思想是将 ReLU 函数与 tanh 函数级联，并引入乘数 ϱ ，使级联函数近似于符号函数。所提出的硬阈值判决函数如下：

$$f_{\varrho}(x) = \max(0, \frac{e^{\varrho x} - e^{-\varrho x}}{e^{\varrho x} + e^{-\varrho x}}). \tag{55}$$

通过将 ReLU 函数与 tanh 函数级联，我们不仅确保基于阈值的决策在输入为负时输出严格为 0，还保证输入为正时，随着 ϱ 的增加，输出逐渐趋近于 1。图 6 绘出了最优符号函数、Sigmoid函数和硬阈值决策函数。图中显示，随着 ϱ 的增大， $f_{\varrho}(\cdot)$ 逐渐趋近于 sng(x)，验证了所提出的硬阈值决策函数的合理性。

备注 *2*： 尽管我们将硬阈值判决组件的应用限制在大规模机器类通信中的非相干传输，但所提出的组件也可用于信号具有特殊稀疏结构的其他场景，例如空间调制系统。此外，所设计的硬阈值决策函数也可用于任何比特级检测器。也就是说，硬阈值判决组件是一个应用范围广泛的即插即用模块。

V. DL-MAMPNET 的实现

A. 参数初始化

在深度学习中，参数初始化对于加快收敛和降低错误率起着关键作用。选择合适的初始化值对于所提出的 DL-mAMPnet 尤为重要，因为 DL-mAMPnet 建立在 AMP 算法之上，因此应保留一些基本特性，以确保性能和可解释性。参数化主要需要考虑三项：可训练矩阵 $\boldsymbol{A}_t$ 和 $\boldsymbol{B}_t$、去噪器参数集 $\boldsymbol{\Theta}_t$以及精炼模块参数 $\boldsymbol{\theta_{RM}} = \{\theta_{FC_1}, \theta_{FC_2}, \theta_{FC_3}, \theta_C\}$。

1) 初始化 $\boldsymbol{A}_t$ 和 $\boldsymbol{B}_t$： 从 (44)-(45) 可以看出，DL-mAMPnet 实现了对 (14)-(15) 中 AMP 算法的推广，其中，匹配滤波器 $(\boldsymbol{S}, \boldsymbol{S}_n^H)$ 在迭代 t 时表现为 $(\boldsymbol{A}_t, \boldsymbol{B}_t)$。然而，这种推广并未强制满足 $\boldsymbol{B}_t = \boldsymbol{A}_t^H$，因此可能无法保留去噪器输入 (19) 的独立高斯特性。根据 [30], 中的分析，当 $\boldsymbol{A}_t = \upsilon_t \boldsymbol{S}$ 且 $\upsilon_t > 0$ 时，所需特性得以保持。因此， $\boldsymbol{A}_t$ 被参数化为 $\upsilon_t \boldsymbol{S}$ ，并且 (44)-(45) 可改写为

$$\tilde{\boldsymbol{X}}_t = \upsilon_t \eta_t(\tilde{\boldsymbol{X}}_{t-1} + \boldsymbol{B}_t \tilde{\boldsymbol{R}}_{t-1}; \boldsymbol{\Theta}_t), \tag{56}$$

$$\tilde{\boldsymbol{R}}_t = \tilde{\boldsymbol{Y}} - \boldsymbol{S}\tilde{\boldsymbol{X}}_t + \frac{\upsilon_t \tilde{\boldsymbol{R}}_{t-1}}{LM} \sum_{j=1}^{2NQM} [\eta_t(\tilde{\boldsymbol{X}}_{t-1} + \boldsymbol{B}_t \tilde{\boldsymbol{R}}_{t-1}; \boldsymbol{\Theta}_t)]_j', \tag{57}$$

其推导过程可参见 [30] ，为简洁起见，此处省略。本文对 $\boldsymbol{B}_t = \tilde{\boldsymbol{S}}^T$ 和 $\upsilon_t = 1$进行初始化，因为这种初始化可以大幅加快训练过程的收敛速度 [30]。

2）初始化 $\boldsymbol{\Theta}_t$： 对于 $\boldsymbol{\theta}_1$，我们将每个元素初始化为 $\log(\frac{Q-\epsilon}{\epsilon})$，也就是说，初始化时假设每个导频序列具有相同的激活概率。这是因为我们对设备活动和发送导频序列索引没有任何先验信息。采用这种均匀初始化后，初始的 $\boldsymbol{\theta}_1$ 与实际值之间的欧几里得距离最小。

例如，考虑一个具有 2 位消息和活动指示器 $\{1,0,0,0\}$ 的设备。如果我们从一个不匹配的独热向量开始，那么欧几里得距离将为 $\sqrt{2}$。如果我们将 $\boldsymbol{\alpha}_n$ 初始化为 $\{\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\}$，那么欧几里得距离将为 $\sqrt{\frac{3}{4}}$。因此，均匀初始化可以加速收敛，因为欧几里得距离较短可能会带来更快的收敛。

$\boldsymbol{\theta}_2$ 的初始值可以根据接收信号强度计算。回想一下，每个导频序列的单位范数为1，且 $\boldsymbol{h}_n \sim \mathcal{CN}(\mathbf{0},\ \beta_n \boldsymbol{I}_M)$，初始 $\boldsymbol{\theta}_2$ 的每个元素大致为 $||\tilde{\boldsymbol{Y}}||_2^2/\sqrt{2K}$。

3) 初始化 $\boldsymbol{\theta}_{\boldsymbol{RM}}$：对于细化模块中的所有参数，我们采用 He 初始化 [34] ，因为已有数学证明表明，它是针对 ReLU 激活函数 [35]的最佳权重初始化策略。

B. 参数训练

1) 训练算法：除了网络结构和参数初始化之外，训练算法也决定了 DL-mAMPnet 的性能。标准训练策略是端到端训练，即按照反向传播规则同时优化所有参数。然而，由于以下原因，端到端训练不适用于 DL-mAMPnet：(i) AMP算法旨在根据 $\boldsymbol{Y}$提供一个使均方误差最小的估计值 $\hat{}\boldsymbol{X}(\boldsymbol{Y})$，即 MSE $\mathbb{E}_{\boldsymbol{XY}}||\hat{}\boldsymbol{X}(\boldsymbol{Y})-\boldsymbol{X}||_2^2$。如果训练 DL-mAMPnet 学习从 $\boldsymbol{Y}$ 到 $\boldsymbol{\alpha}$的直接映射，则可能无法实现 AMP 层的均方误差最优性；(ii) 即使分别训练 AMP 层和细化模块，AMP 层仍可能因过拟合 [36]而轻易收敛到较差的局部最优解。

基于这些原因，我们提出了一种逐层训练策略，其核心思想是将每一层的训练解耦。具体细节见**算法 1**。逐层训练共分为$T+2$ 个阶段。第一阶段，我们训练第一个 AMP 层的可学习参数。然后在第 t 阶段，我们训练前 t 个 AMP 层，并将前 $t-1$ 个 AMP 层的参数固定为前 $t-1$ 个阶段学得的参数。在第$T+1$ 阶段，我们训练整个网络，仅细化模块的参数可学习，而 AMP 层的参数固定为前 T 个阶段学得的参数。最后，在最后一个阶段，将所有参数初始化为前 $T+1$ 个阶段训练得到的参数，然后进行联合训练。

算法 1通过逐层训练策略训练 DL-mAMPnet 的参数**输入：**训练数据集 D_{AMP}，D_{RM}；**输出：**训练后的参数 $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_{t=1}^{T}$和 $\boldsymbol{\theta}_{RM}$；根据 IV-B节初始化参数；

对于 $t = 1$ 到 T **执行**

基于损失函数 (58)，在固定 $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_{t=1}^{t-1}$的情况下学习 $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_t$；
结束循环基于损失函数 (59)，在固定 $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_{t=1}^{T}$ 的情况下学习 $\boldsymbol{\theta}_{RM}$；
基于损失函数 (59) 重新学习 $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_{t=1}^{T}$和 $\boldsymbol{\theta}_{RM}$；**返回** $\{\upsilon_t, \boldsymbol{B}_t, \boldsymbol{\Theta}_t\}_{t=1}^{T}$和 $\boldsymbol{\theta}_{RM}$。

~ ~前 T 个阶段的训练数据集 D_{AMP} 包含 100, 000 对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$，相应的损失函数为均方误差损失

$$\mathcal{L}_t(\tilde{\boldsymbol{Y}}) = ||\tilde{\boldsymbol{X}}_t(\tilde{\boldsymbol{Y}}) - \tilde{\boldsymbol{X}}||_2^2, t = [1, \cdots, T]. \tag{58}$$

~最后 2 个阶段的训练数据集 D_{RM} 包含 100, 000 对 $\boldsymbol{\alpha}$ 和 $\boldsymbol{Y}$，损失函数为二元交叉熵损失

$$\mathcal{L}_t(\tilde{\boldsymbol{Y}}) = \frac{1}{NQ}\sum_{i=1}^{NQ}\left(\boldsymbol{\alpha}(\tilde{\boldsymbol{Y}})_i \log \boldsymbol{\alpha}_i + (1 - \boldsymbol{\alpha}(\tilde{\boldsymbol{Y}})_i)\log(1 - \boldsymbol{\alpha}_i)\right), t = [T+1, T+2]. \tag{59}$$

DL-mAMPnet 使用 Adam 优化器，通过训练数据集逐轮进行训练；每轮训练中，将整个训练数据集打乱并划分为大小为 500 的批次。[1]

2）训练数据集：训练数据集按如下方式合成：(i) 生成$\boldsymbol{\alpha}_n$：从 N 个设备中随机选取 K 个活跃设备。然后，为每个活跃设备随机分配一个 Q维独热向量，并为每个非活跃设备分配一个 Q维零向量；(ii) 生成 $\boldsymbol{X}_n$：首先根据式（1）生成设备 n的上行信道，即 $\boldsymbol{h}_n$。然后，通过将 $\boldsymbol{h}_n$ 与 $\boldsymbol{\alpha}_n$相乘得到 $\boldsymbol{X}_n$；(iii) 生成$\boldsymbol{Y}$：通过从零均值、给定方差的复高斯分布中采样，生成导频序列 $\boldsymbol{S}_n$。给定 $\boldsymbol{X}_n$ 和 $\boldsymbol{S}_n$后，可根据式（6）直接得到 $\boldsymbol{Y}$。

[1]需要指出的是，每个阶段的训练轮次和学习率都不相同，这些参数是在 V节中通过经验确定的。

VI. 仿真结果

本节通过大量仿真验证所提算法的有效性。除非另有说明，仿真设置如下。为便于说明，我们考虑一个包含$N = 100$ 个设备的大规模机器类通信系统，尽管所提算法可应用于规模大得多的系统。在每个相干块中，每个设备以 ϵ =0.1的概率独立接入基站。设备 n 的大尺度衰落系数为 $\beta_n = 128.1 - 36.7\log_{10}(d_n)$ dB，其中 d_n 为设备 n 与基站之间的距离，且该距离在 [0.05, 1] km范围内服从均匀分布。每个设备的小尺度衰落系数均服从零均值、单位方差的独立同分布多元复高斯分布。假设基站处加性高斯白噪声的功率谱密度为 -169 dBm/Hz [8]，无线信道带宽为 1 MHz。

DL-mAMPnet 中 AMP 层的数量设为 $T = 4$。每个训练阶段的训练轮数和学习率分别设为 $\{2,000, 1,500, 1,000, 1,000, 1,500, 5,000\}$ 和 $\{2 \times 10^{-5}, 2 \times 10^{-5}, 2 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}, 1 \times 10^{-5}\}$。[2] 我们使用 80,000 个训练样本训练 DL-mAMPnet，并使用 20,000 个数据样本进行测试；这些样本分别从前 4 个阶段的D_{AMP} 和最后 2 个阶段的 D_{RM} 中随机抽取。DL-mAMPnet 在一台配备 Nvidia GeForce GTX 1080 Ti 显卡的 x86个人计算机上进行训练和测试，后端采用 PyTorch 1.1.0。采用具有 $T_{AMP} = 50$ 次迭代的传统的基于AMP的算法和具有 $T_{Cov} = 50$ 次迭代的基于协方差的方法 [16] 作为基准方法，并在同一数据集上进行评估。此外，采用符号错误率作为性能指标：SER $= \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}(\hat{\boldsymbol{\alpha}}_n \neq \boldsymbol{\alpha}_n)$，其中 $\hat{\boldsymbol{\alpha}}_n$ 和 $\boldsymbol{\alpha}_n$ 分别表示设备 n 的导频序列活动估计值及其真实值。

—

*A. DL-mAMPnet*的性能

图7展示了不同 M取值下，SER 随 L 的变化情况。可以观察到，随着 L 和 M 增大，DL-mAMPnet 和基于 AMP 的算法的 SER 均有所下降。尽管当 L 较小时，基于协方差的算法的 SER 最低，但当 L超过某个值（例如，当 $M = 16$时的 $L = 40$ ）后，其性能便趋于饱和。这主要是由于该算法的次优性

[2]所有参数均通过通用工作流程以经验方式确定：训练从相对较小的数值开始，逐步增大这些数值，直至学习性能无法进一步提升。

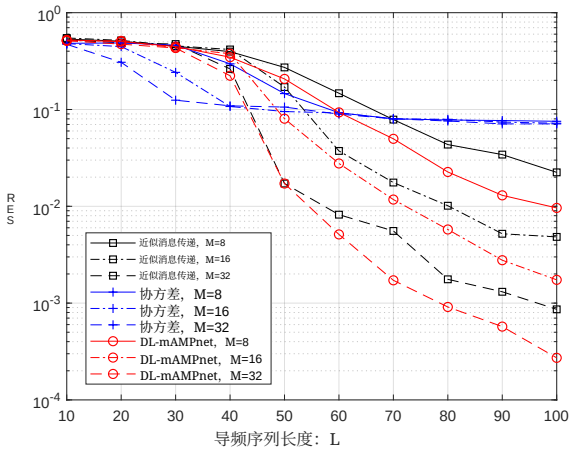


图7. 符号错误率性能随导频序列长度 L 的变化（ $J=1$ 位）。

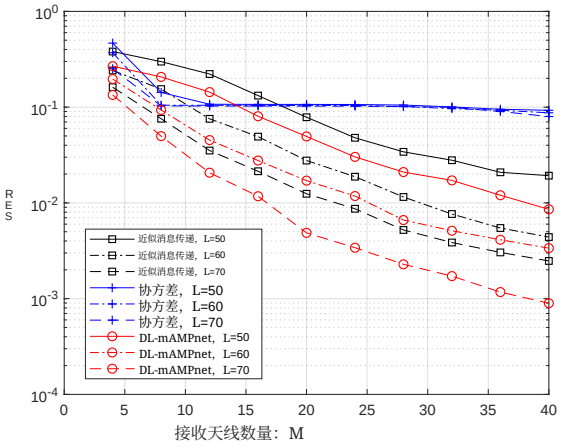


图8. 符号错误率性能随接收天线数量 M 的变化（ $J=1$ 位）。

固定阈值。与此同时，所提出的 DL-mAMPnet 明显优于基于 AMP 的算法，优势显著。例如，当 L 大于 70 时，所提出的 DL-mAMPnet 相较于基于 AMP 的算法可获得超过 10 的导频长度增益，这表明所提出的 DL-mAMPnet 能够缩短所需导频序列长度，降低导频设计难度，并适应快速变化信道。此外，尽管对于任意 M，DL-mAMPnet 和基于 AMP 的算法的 SER 均随 L增大而下降，但前者下降得更快

[3]由（37）可知，阈值是可变的，并与信号功率和接收天线数量等系统参数有关，而基于协方差的算法采用固定阈值。由于没有具体的方法来设计这种固定阈值，本文通过经验将基于协方差的算法的阈值设为 $\beta_n/2$ 。

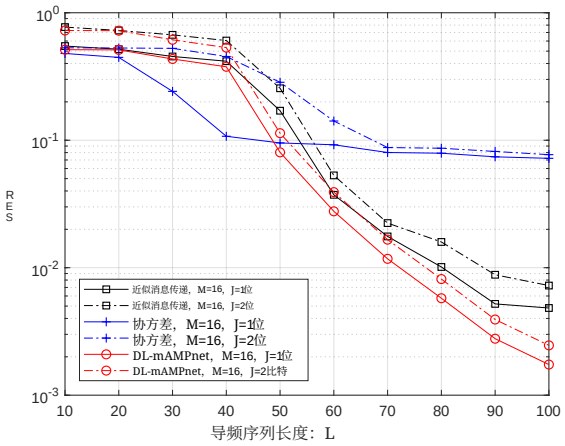


图9. 不同传输消息长度 J下，符号错误率性能随导频序列长度 L 的变化。

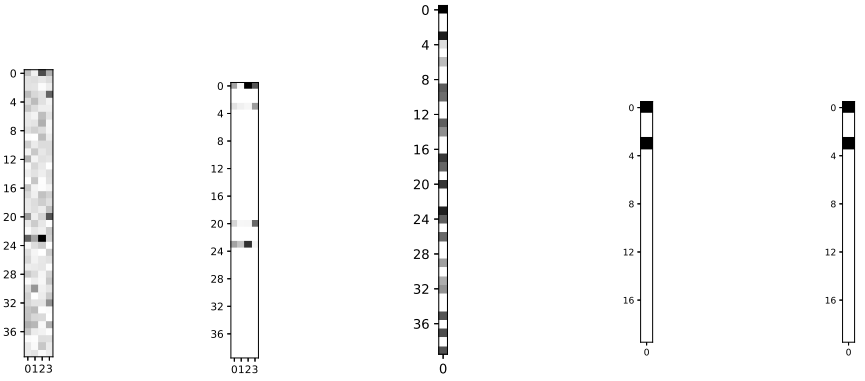
当 M 为32时，与 M 为8时相比，这表明增加接收天线数量可以进一步缩短所需的导频序列长度。

图8展示了不同 L取值下，符号错误率随 M 的变化。我们观察到，对于 DL-mAMPnet 和基于 AMP 的算法，随着 M 增加，符号错误率会有效降低；而对于基于协方差的算法，符号错误率存在误差地板。此外，DL-mAMPnet 达到与基于 AMP 的算法相同性能所需的接收天线更少，这表明所提出的 DL-mAMPnet 可以降低对接收天线的需求，从而降低部署成本和能耗。

图9绘制了两种不同传输消息长度下，即 $J = 1$ 比特和 $J = 2$ 比特时，符号错误率随 L的变化。接收天线数量为 $M = 16$。可以看出，随着传输消息长度增加，三种算法的符号错误率均会上升，这表明传输更多消息时，两种算法的性能都会下降。值得注意的是，随着消息长度增加，所提出的 DL-mAMPnet 与另外两种算法之间的性能差距也会扩大，这表明 DL-mAMPnet 具有处理较大数据包的潜力。

B. DL-mAMPnet 的可视化

为了进一步了解所提出的 DL-mAMPnet，我们展示了一个训练良好的 DL-mAMPnet 各组件输出的可视化结果。为清晰起见，我们仅展示 $N = 10$ 个设备以 $\epsilon = 0.1$ 激活概率发送 1 比特消息、导频



(a) (b) (c) (d) (e)

图 10. 训练良好的 DL-mAMPnet 的可视化。(a) $\tilde{\boldsymbol{X}}_T$，AMP 层的输出；(b) $\tilde{\boldsymbol{X}}$ 的真实值；(c) Mp(Mup($\boldsymbol{o}$))，最大池化-最大反池化过程的输出；(d) $\hat{\boldsymbol{\alpha}}$，细化模块的输出；(e) $\boldsymbol{\alpha}$ 的真实值。

序列长度为 $M = 2$ 且接收天线数为 $M = 2$ 的情况。为了进行可视化，我们将各组件的输出转换为反向灰度图像。具体而言，将每个输出矩阵的元素归一化到 $[0, 1]$，范围内，其中 0 和 1 分别以白色和黑色表示。需要说明的是，我们对 $\tilde{\boldsymbol{X}}$ 和 $\tilde{\boldsymbol{X}}_T$ 取绝对值，以便更直观地展示信号强度差异。

AMP 层的输出及其真实值分别如图 10(a) 和图 10(b)所示。可以看出，$\tilde{\boldsymbol{X}}$ 的非零行均被正确恢复，为后续细化过程奠定了基础。随后，图 10(c) 展示了最大池化-最大反池化过程的输出，其中相邻两行中较大的值被保留，另一行变为 0，这证明了最大池化-最大反池化过程在确保设备级稀疏性方面的有效性。图 10(d) 和图 10(e) 分别展示了$\hat{\boldsymbol{\alpha}}$ 和 $\boldsymbol{\alpha}$的可视化结果，可以看出，训练良好的 DL-mAMPnet 能够完美估计导频序列活动。此外，从图 10(c) 和图 10(e) 可以观察到，图 10(c) 中正确保留了导频序列活动（第 1、4、21 和 24 行），这表明所提出的软阈值去噪组件是有效的。

C. 计算复杂度分析

最后，我们分析传统的基于AMP的算法和 DL-mAMPnet 的计算复杂度。

对于传统的基于AMP的算法，计算复杂度主要来自式（14）–（15）中的矩阵乘法[8]。由于 $\boldsymbol{S}_n^H \in \mathbb{C}^{Q\times L}$、 $\boldsymbol{R}_t \in \mathbb{C}^{L\times M}$、 $\boldsymbol{S} \in \mathbb{C}^{L\times NQ}$ 和 $\boldsymbol{X}_{t+1} \in \mathbb{C}^{NQ\times M}$， N 个设备经过 T_{AMP} 次迭代的计算复杂度为$\mathcal{O}(4T_{AMP}(NQLM + NQLM)) = \mathcal{O}(8T_{AMP}NQLM)$，其中，比例常数“4”是因为一次复数乘法需要 4 次实数乘法，前面的“$NQLM$”来自 N个设备中 $\boldsymbol{S}_n^H$与 $\boldsymbol{R}_t$之间的乘法，后面的“$NQLM$”来自 $\boldsymbol{S}$ 与 $\boldsymbol{X}_{t+1}$之间的乘法。迭代过程结束后，基于 AMP 的算法需要执行元素选择操作（即式（35）），其计算复杂度为 $\mathcal{O}(4NQM)$；还需要进行阈值计算（即式（37）），其计算复杂度为$\mathcal{O}(4NQM)$。综合考虑所有操作，基于 AMP 的算法的计算复杂度为 $\mathcal{O}(8T_{AMP}NQLM)$。

对于所提出的方法 DL-mAMPnet，我们重点分析在线实现的计算复杂度。AMP 层的计算复杂度来自矩阵乘法 $\boldsymbol{B}_t\tilde{\boldsymbol{R}}_{t-1}$和 $\boldsymbol{A}_t\tilde{\boldsymbol{X}}_t$，其为 $\mathcal{O}(8T_{DL}NQLM^2)$，其中 T_{DL}表示 AMP 层的数量。对于细化模块，计算复杂度主要来自全连接层和卷积层。对于一个具有 N_{l-1} 个输入和 N_1 个输出的全连接层，其计算复杂度为 $\mathcal{O}(N_{l-1}N_1)$。对于一个具有 $H \times W$ 个输入和 $H_f \times W_f$个滤波器的卷积层，其计算复杂度可表示为 $\mathcal{O}(HWH_fW_f)$。因此，细化模块的总计算复杂度为 $\mathcal{O}(4N^2Q^2M)$。 consequently，DL-mAMPnet 的计算复杂度为 $\mathcal{O}(8T_{DL}NQLM^2 + 4N^2Q^2M)$。

综上所述，与基于 AMP 的算法相比，所提出的 DL-mAMPnet 似乎能够以更高的计算复杂度为代价实现更好的性能。然而，如图7至图9所示，具有 $T_{DL} = 4$ AMP层的 DL-mAMPnet 优于经过 $T_{AMP} = 50$ 次迭代的基于 AMP 的算法，这表明所提出的 DL-mAMPnet 可能只需较低的计算复杂度，就能达到与基于 AMP 的算法相同的符号错误率性能。

VII. 结论

本文提出了一种新颖的基于深度学习的算法，称为 DL-mAMPnet，用于单阶段非相干方案下的大规模机器类通信中的设备活动与数据联合检测。DL-mAMPnet 中加入了可训练参数，以补偿传统 AMP 算法中独立同分布假设导致的不准确性。此外，还进一步设计了一个细化模块

，通过利用相关稀疏模式来提升符号错误率性能，并确保设备级稀疏性。所提出的方法适用于海量用户间歇性传输小数据包的场景，例如智能家居和工业控制。未来，我们将研究导频序列设计方案，以保持正交性并减轻设备间干扰。

附录A

MMSE去噪器（21）的推导

为便于推导条件概率 $P_{\boldsymbol{X}_n|\boldsymbol{Z}_n}$，我们假设 $\boldsymbol{x}_n^q$ 彼此独立，因此有

$$P_{\boldsymbol{x}_n^q} = \left(1-\frac{\epsilon}{Q}\right)\delta + \frac{\epsilon}{Q}\frac{\exp(-\boldsymbol{x}_n^{q\,H}(\beta_n\boldsymbol{I}_M)^{-1}\boldsymbol{x}_n^q)}{\pi^M|\beta_n\boldsymbol{I}_M|}. \tag{60}$$

根据（20），给定 $\boldsymbol{x}_n^q$ 时观测到 $\boldsymbol{z}_n^q$ 的似然为

$$P_{\boldsymbol{z}_n^q|\boldsymbol{x}_n^q} = \frac{\exp(-(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q)^H\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q))}{\pi^M|\boldsymbol{\Sigma}|}. \tag{61}$$

将 k 记为比例常数， $P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q}$ 可利用贝叶斯公式计算如下

$$\begin{aligned}
&P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q} = kP_{\boldsymbol{z}_n^q|\boldsymbol{x}_n^q}P_{\boldsymbol{x}_n^q}\\
&= k\left((1-\frac{\epsilon}{Q})\delta + \frac{\epsilon}{Q}\frac{\exp(-\boldsymbol{x}_n^{q\,H}(\beta_n\boldsymbol{I}_M)^{-1}\boldsymbol{x}_n^q)}{\pi^M|\beta_n\boldsymbol{I}_M|}\right)\left(\frac{\exp(-(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q)^H\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q))}{\pi^M|\boldsymbol{\Sigma}|}\right)\\
&= k\left((1-\frac{\epsilon}{Q})\frac{\exp(-\boldsymbol{z}_n^{q\,H}\boldsymbol{\Sigma}^{-1}\boldsymbol{z}_n^q)}{\pi^M|\boldsymbol{\Sigma}|}\delta + \frac{\epsilon}{Q}\frac{\exp(-\boldsymbol{x}_n^{q\,H}(\beta_n\boldsymbol{I}_M)^{-1}\boldsymbol{x}_n^q-(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q)^H\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q))}{\pi^{2M}|\beta_n\boldsymbol{I}_M||\boldsymbol{\Sigma}|}\right).
\end{aligned} \tag{62}$$

注意

$$\boldsymbol{x}_n^{q\,H}(\beta_n\boldsymbol{I}_M)^{-1}\boldsymbol{x}_n^q + (\boldsymbol{z}_n^q-\boldsymbol{x}_n^q)^H\boldsymbol{\Sigma}^{-1}(\boldsymbol{z}_n^q-\boldsymbol{x}_n^q) = (\boldsymbol{x}_n^q-\boldsymbol{\zeta})^H\boldsymbol{\Xi}^{-1}(\boldsymbol{x}_n^q-\boldsymbol{\zeta}) + \boldsymbol{z}_n^{q\,H}\boldsymbol{\Delta}^{-1}\boldsymbol{z}_n^q,$$

其中 $\boldsymbol{\Xi} = (\frac{1}{\beta_n}\boldsymbol{I}_M + \boldsymbol{\Sigma}^{-1})$、$\boldsymbol{\zeta} = \boldsymbol{\Xi}\boldsymbol{\Sigma}^{-1}\boldsymbol{z}_n^q$和 $\boldsymbol{\Delta} = \beta_n\boldsymbol{I}_M + \boldsymbol{\Sigma}$，(62)可改写为 s

$$\begin{aligned}
&P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q}\\
&= k\left((1-\frac{\epsilon}{Q})\frac{\exp(-\boldsymbol{z}_n^{q\,H}\boldsymbol{\Sigma}^{-1}\boldsymbol{z}_n^q)}{\pi^M|\boldsymbol{\Sigma}|}\delta + \frac{\epsilon}{Q}\frac{\exp\left(-(\boldsymbol{x}_n^q-\boldsymbol{\zeta})^H\boldsymbol{\Xi}^{-1}(\boldsymbol{x}_n^q-\boldsymbol{\zeta})-\boldsymbol{z}_n^{q\,H}\boldsymbol{\Delta}^{-1}\boldsymbol{z}_n^q\right)}{\pi^{2M}|\beta_n\boldsymbol{I}_M||\boldsymbol{\Sigma}|}\right).
\end{aligned} \tag{63}$$

由于 $\int P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q} d\boldsymbol{x}_n^q = 1$, k 可通过对(63)积分消去得到。因此，我们有

$$\begin{aligned} k &= \left((1-\frac{\epsilon}{Q}) \frac{\exp(-\boldsymbol{z}_n^{qH} \boldsymbol{\Sigma}^{-1} \boldsymbol{z}_n^q)}{\pi^M |\boldsymbol{\Sigma}|} + \frac{\epsilon}{Q} \frac{\exp\left(-\boldsymbol{z}_n^{qH} \boldsymbol{\Delta}^{-1} \boldsymbol{z}_n^q\right) |\boldsymbol{\Xi}|}{\pi^M |\beta_n \boldsymbol{I}_M| |\boldsymbol{\Sigma}|} \right)^{-1} \\ &\overset{(a)}{=} \left((1-\frac{\epsilon}{Q}) \frac{\exp(-\boldsymbol{z}_n^{qH} \boldsymbol{\Sigma}^{-1} \boldsymbol{z}_n^q)}{\pi^M |\boldsymbol{\Sigma}|} + \frac{\epsilon}{Q} \frac{\exp\left(-\boldsymbol{z}_n^{qH} \boldsymbol{\Delta}^{-1} \boldsymbol{z}_n^q\right)}{\pi^M |\boldsymbol{\Delta}|} \right)^{-1}, \end{aligned} \tag{64}$$

其中，(a)成立是因为 $|\frac{1}{\beta_n} \boldsymbol{I}_M + \boldsymbol{\Sigma}^{-1}| = |\beta_n \boldsymbol{I}_M||\boldsymbol{\Sigma}|/|\beta_n \boldsymbol{I}_M + \boldsymbol{\Sigma}|$。

将(64)代入(62)， $P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q}$ 可确定为

$$P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q} = \frac{e^{-(\boldsymbol{x}_n^q - \boldsymbol{\zeta})^H \boldsymbol{\Xi}^{-1} (\boldsymbol{x}_n^q - \boldsymbol{\zeta})} \epsilon |\boldsymbol{\Sigma}| + (Q-\epsilon) e^{-\boldsymbol{z}_n^{qH} (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Delta}^{-1}) \boldsymbol{z}_n^q} \pi^M |\boldsymbol{\Xi}| |\boldsymbol{\Delta}| \delta}{\epsilon \pi^M |\boldsymbol{\Xi}| |\boldsymbol{\Sigma}| + (Q-\epsilon) e^{-\boldsymbol{z}_n^{qH} (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Delta}^{-1}) \boldsymbol{z}_n^q} \pi^M |\boldsymbol{\Xi}| |\boldsymbol{\Delta}|}. \tag{65}$$

因此，条件期望 $\mathbb{E}\{\boldsymbol{x}_n^q | \boldsymbol{z}_n^q\}$ 为

$$\begin{aligned} \mathbb{E}\{\boldsymbol{x}_n^q | \boldsymbol{z}_n^q\} &= \int \boldsymbol{x}_n^q P_{\boldsymbol{x}_n^q|\boldsymbol{z}_n^q} \, d\boldsymbol{x}_n^q = \frac{\boldsymbol{\zeta} \epsilon |\boldsymbol{\Sigma}|}{\epsilon |\boldsymbol{\Sigma}| + (Q-\epsilon) e^{-\boldsymbol{z}_n^{qH} (\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Delta}^{-1}) \boldsymbol{z}_n^q} |\boldsymbol{\Delta}|} \\ &= \frac{\beta_n (\beta_n \boldsymbol{I}_M + \boldsymbol{\Sigma})^{-1} \boldsymbol{z}_n^q}{1 + \frac{Q-\epsilon}{\epsilon} |\boldsymbol{I}_M + \beta_n \boldsymbol{\Sigma}^{-1}| e^{-\boldsymbol{z}_n^{qH} (\boldsymbol{\Sigma}^{-1} - (\boldsymbol{\Sigma} + \beta_n \boldsymbol{I}_M)^{-1}) \boldsymbol{z}_n^q}}. \end{aligned} \tag{66}$$

(21)-(25)中的MMSE最优去噪器可通过对(66)进行简单的数学变换直接得到。

参考文献

[1] N. H. Mahmood等，《面向6G的关键和大规模机器类型通信白皮书》，*arXiv*预印本，arXiv:2004.14146，2020。[2] X. Chen、D. W. K. Ng、W. Yu、E. G. Larsson、N. Al-Dhahir和R. Schober，《面向5G及后续演进的海量接入》，*IEEEJ. Sel. Areas Commun.*，第39卷，第3期，第615–637页，2021年3月。[3] L. Liu、E. G. Larsson、W. Yu、P. Popovski、C. Stefanovic和E. de Carvalho，《免授权大规模连接的稀疏信号处理：物联网随机接入协议的未来范式》，*IEEE Signal Process.Mag.*，第35卷，第5期，第88–99页，2018年9月。[4] C. Bockelmann、N. Pratas、H. Nikopour、K. Au、T. Svensson、C. Stefanovic、P. Popovski和A. Dekorsy，《5G中的大规模机器类型通信：物理层和MAC层解决方案》，*IEEE Commun.Mag.*，第54卷，第9期，第59–65页，2016年9月。[5] M. B. Shahab、R. Abbas、M. Shirvanimoghaddam和S. J. Johnson，《物联网免授权非正交多址接入：综述》，*IEEE Commun. Surveys Tuts.*，第22卷，第3期，第1805–1838页，2020年5月。[6] E. Bjornson、E. de Carvalho、J. H. Sorensen、E. G. Larsson和P. Popovski，《拥挤大规模MIMO系统中用于导频分配的随机接入协议》，*IEEE Trans. Wireless Commun.*，第16卷，第4期，第2220–2234页，2017年4月。[7] Z. Chen、F. Sohrabi和W. Yu，《大规模连接的稀疏活动检测》，*IEEE Trans. Signal Process.*，第66卷，第7期，第1890–1904页，2018年4月。[8] L. Liu和W. Yu，《大规模MIMO下的大规模连接——第一部分：设备活动检测与信道估计》，*IEEE Trans. Signal Process.*，第66卷，第11期，第2933–2946页，2018年6月。[9] X. Zhang、F. Labeau、L. Hao和J. Liu，《MTC通信中基于贝叶斯学习方法的联合活跃用户检测和信道估计》，*IEEE Trans. Veh. Technol.*，第70卷，第6期，第6222–6226页，2021年6月。[10] J. W. Choi、B. Shim和S.-H. Chang，《通过压缩感知缩减大规模MIMO系统的下行导频》，*IEEE Commun. Lett.*，第19卷，第11期，第1889–1892页，2015年11月。[11] T. V. Luong、Y. Ko、N. A. Vien、M. Matthaiou和H. Q. Ngo，《非相干多载波MU-SIMO系统的深度能量自编码器》，*IEEE Trans. Wireless Commun.*，第19卷，第6期，第3952–3962页，2020年6月。

[12] S. Xue、Y. Ma和N. Yi，《衰落信道中用于上行MU-SIMO联合发射机与非相干接收机设计的端到端学习》，IEEE Trans. Wireless Commun.，第20卷，第9期，第5531–5542页，2021年9月。[13] K. Senel和E. G. Larsson，《使用AMP进行大规模MIMO中的设备活动和嵌入式信息比特检测》，载于IEEE GLOBECOM会议论文集，2017年，第1–6页。[14] K. Senel和E. G. Larsson，《利用机器学习算法在支持mMTC的大规模MIMO中进行联合用户活动与非相干数据检测》，载于WSA会议论文集，2018年，第1–6页。[15] K. Senel和E. G. Larsson，《免授权大规模MTC支持的大规模MIMO：一种压缩感知方法》，IEEE Trans. Commun.，第66卷，第12期，第6164–6175页，2018年12月。[16] Z. Chen、F. Sohrabi、Y.-F. Liu和W. Yu，《基于协方差的采用大规模MIMO的大规模随机接入联合活动与数据检测》，载于IEEE ICC会议论文集，2019年，第1–6页。[17] X. Shen、J. Gao、W. Wu、M. Li、C. Zhou和W. Zhuang，《面向6G的整体网络虚拟化与普适网络智能》，IEEE Commun. Surveys Tuts.，第24卷，第1期，第1–30页，2022年第一季度。[18] X. Shen等，《基于AI辅助网络切片的下一代无线网络》，IEEE Open J. Veh. Technol.，第1卷，第45–66页，2020年1月。[19] W. Wu、N. Cheng、N. Zhang、P. Yang、W. Zhuang和X. Shen，《通过相关多臂老虎机学习实现快速毫米波波束对齐》，IEEE Trans. Wireless Commun.，第18卷，第12期，第5894–5908页，2019年12月。[20] H. He、C. Wen、S. Jin和G. Y. Li，《面向MIMO检测的模型驱动深度学习》，IEEE. Trans. Signal Process.，第68卷，第1期，第1702–1715页，2020年2月。[21] M. Soltani、V. Pourahmadi、A. Mirzaei和H. Sheikhzadeh，《基于深度学习的信道估计》，IEEE Commun. Lett.，第23卷，第4期，第652–655页，2019年4月。[22] Z. Ma、W. Wu、M. Jian、F. Gao和X. Shen，《免授权NOMA的联合星座设计与多用户检测》，IEEE Trans. Wireless Commun.，第21卷，第3期，第1973–1988页，2022年3月。[23] J. R. Hershey、J. Le Roux和F. Weninger，《深度展开：新型深度架构的模型驱动灵感》，arXiv预印本，arXiv:1409.2574，2014年。[24] H. He、S. Jin、C.-K. Wen、F. Gao、G. Y. Li和Z. Xu，《面向物理层通信的模型驱动深度学习》，IEEE Wireless Commun.，第26卷，第5期，第77–83页，2019年10月。[25] J. Kim、W. Chang、B. Jung、D. Baron和J. C. Ye，《用于联合稀疏恢复的置信传播》，arXiv预印本，arXiv:1102.3289，2011年。[26] J. A. Tropp，《同时稀疏逼近算法。第二部分：凸松弛》，Signal Processing，第86卷，第3期，第589–602页，2006年。[27] D. L. Donoho、A. Maleki和A. Montanari，《用于压缩感知的消息传递算法》，Proc. Nat. Acad. Sci. USA，第106卷，第45期，第18914–18919页，2009年11月。[28] S. Rangan，《采用随机线性混合进行估计的广义近似消息传递》，载于IEEE ISIT会议论文集，2011年，第2168–2172页。[29] M. Bayati和A. Montanari，《稠密图上的消息传递动态及其在压缩感知中的应用》，IEEE Trans. Inf. Theory，第57卷，第2期，第764–785页，2011年2月。[30] M. Borgerding、P. Schniter和S. Rangan，《用于稀疏线性逆问题的受近似消息传递启发的深度网络》，IEEE Trans. Signal Process.，第65卷，第16期，第4293–4308页，2017年8月。[31] D. L. Donoho，《通过软阈值处理去噪》，IEEE Trans. Inf. Theory，第41卷，第3期，第613–627页，1995年5月。[32] D. Ciregan、U. Meier和J. Schmidhuber，《用于图像分类的多列深度神经网络》，载于IEEE CVPR会议论文集，2012年，第3642–3649页。[33] X. Gao等，《ComNet：正交频分复用接收机中深度学习与专家知识的结合》，IEEE Commun. Lett.，第22卷，第12期，第2627–2630页，2018年12月。[34] K. He、X. Zhang、S. Ren和J. Sun，《深入研究整流器：在ImageNet分类中超越人类水平的性能》，载于IEEE ICCV会议论文集，2015年，第1026–1034页。[35] S. K. Kumar，《关于深度神经网络中的权重初始化》，arXiv预印本，arXiv:1704.08863，2017年。[36] C. Metzler、A. Mousavi和R. Baraniuk，《学习型D-AMP：基于原理的神经网络压缩图像恢复》，载于NIPS会议论文集，2017年，第1772–1783页。