A Novel Multi-Task Learning Empowered Codebook Design for Downlink SCMA Networks

Qu Luo, *Student Member, IEEE,* Zilong Liu, *Senior Member, IEEE,* Gaojie Chen, *Senior Member, IEEE,*
Yi Ma, *Senior Member, IEEE,* Pei Xiao, *Senior Member, IEEE.*

Abstract

Sparse code multiple access (SCMA) is a promising code-domain non-orthogonal multiple access (NOMA) scheme for the enabling of massive machine-type communication. In SCMA, the design of good sparse codebooks and efficient multiuser decoding have attracted tremendous research attention in the past few years. This paper aims to leverage deep learning to jointly design the downlink SCMA encoder and decoder with the aid of autoencoder. We introduce a novel end-to-end learning based SCMA (E2E-SCMA) design framework, under which improved sparse codebooks and low-complexity decoder are obtained. Compared to conventional SCMA schemes, our numerical results show that the proposed E2E-SCMA leads to significant improvements in terms of error rate and computational complexity.

Index Terms

SCMA, codebook design, deep neural network, autoencoder, multi-task learning.

I. Introduction

THE wireless networks are rapidly evolving towards a paradigm shift from connecting people to networking everything. A pressing challenge of future wireless network design is how to develop a highly efficient multiple access scheme to meet various stringent requirements such as lower access latency, and higher spectral efficiency. A disruptive technique for addressing such a challenge is called non-orthogonal multiple access (NOMA). In a NOMA system, multiple users are able to communicate simultaneously to achieve overloading factor larger than 1. Existing NOMA techniques can be largely categorized into two classes: power-domain NOMA and code-domain NOMA (CD-NOMA) [1], [2]. In this paper, we focus on an emerging CD-NOMA scheme called sparse code multiple access (SCMA) in which multiple users are separated by adopting different sparse codebooks [3], [4]. Over the past decade, SCMA has attracted tremendous research attention from both academia and industry [5]–[8].

In SCMA, two fundamental research problems are the design of good sparse codebooks and efficient multi-user decoding [5], [6], [9], [10]. Existing known SCMA codebook constructions mostly follow a multi-stage sub-optimal design for rapid generation [5], [6], [10], albeit it is unclear how far the obtained SCMA codebooks are from the optimal ones. By taking advantage of the codebook sparsity, low-complexity MPA has been developed for SCMA decoding. For a downlink SCMA system where multiple user devices (e.g., sensors, tablets, machines) are constrained by their limited computation capability and battery life, however, the current MPA may not be affordable, especially when a large number of MPA iterations is needed [9], [11].

In recent years, deep learning (DL) has been extensively studied in wireless networks, thanks to its capability in solving very complicated optimization problem [12]. A comprehensive introduction on autoencoder for end-to-end communication system was contributed by O'shea and Hoydis in [12].

Qu Luo, Gaojie Chen, Yi Ma and Pei Xiao are with 5G & 6G Innovation Centre, University of Surrey, UK, email:{q.u.luo, gaojie.chen, y.ma, p.xiao}@surrey.ac.uk. Zilong Liu is with the School of Computer Science and Electronics Engineering, University of Essex, UK. email: zilong.liu@essex.ac.uk. This work was supported in part by the UK Engineering and Physical Sciences Research Council under Grant EP/P03456X/1 and the Key Project of Science and Technology of Hainan (N0. ZDKJ2019003).

Following [12], a denoising autoencoder (DAE) for SCMA was reported in [13]. The core idea of [13] is to model the entire SCMA system as a DAE by implementing both the encoder and decoder with fully connected neural networks (NNs). Subsequently, a similar structure was studied in [11], [14] by jointly considering the sparse and dense mapping of CD-NOMA. It is noted that [11], [13], [14] considered the decoder as a single learning task implemented with fully connected layers. However, the bit error performances of these systems may not beat an SCMA system with the aforementioned sparse codebooks that are obtained from a multi-stage sub-optimal design. Very recently, a deep neural network (DNN) with multi-task structure was proposed in [15] for SCMA detection. However, [15] has not touched the sparse codebook design with the aid of DNN, and hence a good error rate performance may not be guaranteed.

In this letter, we introduce a novel multi-task learning empowered end-to-end SCMA (E2E-SCMA) design framework. The main novelty of this work stems from the proposed architecture of E2E-SCMA and the unique training scheme. Building upon a new SCMA mapping design with linear encoding, we first propose an efficient SCMA encoder, which can reduce the depth of the network and thereby helping prevent the gradient from vanishing. Unlike existing works [11], [13], [14], where the decoding is conducted by viewing J users as a single learning task, we view each user as a single learning task and then design the decoder in a task-specific fashion. The advantages of using the multi-task learning structure are twofold: 1) it can improve learning efficiency and reduce over-fitting [16]; 2) it can avoid the curse of dimensionality while using one-hot encoding. Specifically, for an multi-task learning structure of J tasks, if each task has a M-dimensional input vector, the corresponding input dimension of single task learning structure will increase to M^J. Finally, we propose to train the E2E-SCMA in a range of signal-to-noise ratios (SNRs) instead of over a fixed SNR. Consequently, this enables the proposed E2E-SCMA to work over a wide range of SNR values with a low error rate performance. The remainder of the letter is organized as follows. Section II briefly describes the system model of SCMA. We present the proposed E2E-SCMA framework in Section III. The numerical results and conclusion are presented in Sections IV and V, respectively.

II. System Model

In this paper, we consider a downlink SCMA system with J users communicating over the K orthogonal resources, where $J > K$. Let us define the overloading factor as $\lambda = \frac{J}{K} > 1$. At the transmitter side, the SCMA encoder maps $\log_2(M)$ binary bits toa length-K codeword drawn from codebook $\mathcal{X}_j \in \mathbb{C}^K$ with size M. The mapping process is defined as $f_j : \mathbb{B}^{\log_2 M} \rightarrow \mathcal{X}_j \in \mathbb{C}^K$, where $\mathcal{X}_j = \{\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, \ldots, \mathbf{x}_{j,m}\}$ is the codebook set for the jth user with cardinality of M. All the K-dimensional complex codewords of each SCMA codebook are sparse vectors with N non-zero elements[1] and $N < K$. Let $\mathbf{c}_j$ be a length-N vector drawn from $\mathcal{C}_j \subset \mathbb{C}^N$, where $\mathcal{C}_j$ is obtained by removing all the zero elements in $\mathcal{X}_j$. We further define the mapping from $\mathbb{B}^{\log_2 M}$ to $\mathcal{C}_j$ as

$$g_j : \mathbb{B}^{\log_2 M \times 1} \mapsto \boldsymbol{\mathcal{C}}_j, \quad \text{i.e., } \mathbf{c}_j = g_j(\mathbf{b}_j), \tag{1}$$

where $\mathbf{b}_j = [b_{j,1}, b_{j,2}, \ldots, b_{j,\log_2 M}]^T \in \{1, -1\}^{\log_2 M}$ stands for jth user's instantaneous input binary message vector. By collecting all the $\mathbf{b}_j$ according to their corresponding integer values in ascending order, we form a $\log_2(M) \times M$ binary matrix $\mathbf{B}$. For example, when $M = 4$, we have

$$\mathbf{B} = \begin{bmatrix} -1 & +1 & -1 & +1 \\ -1 & -1 & +1 & +1 \end{bmatrix}. \tag{2}$$

Thus, the corresponding SCMA mapping f_j can be expressed as

$$f_j :\equiv \mathbf{V}_j g_j, \quad \text{i.e., } \mathbf{x}_j = \mathbf{V}_j g_j(\mathbf{b}_j), \tag{3}$$

where $\mathbf{V}_j \in \mathbb{B}^{K \times N}$ is an mapping matrix that maps the N-dimensional vector to a K-dimensional sparse SCMA codeword. The sparse structure of the J SCMA codebooks can be represented by the indicator (sparse) matrix $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_J] \subset \mathbb{B}^{K \times J}$ where $\mathbf{f}_j = \text{diag}(\mathbf{V}_j \mathbf{V}_j^T)$.

[1]For user j, the N non-zero element positions remain unchanged from one codeword to another.

For a fixed $\mathbf{V}_j$, the task of SCMA codebook design is to find the J mapping functions $g_j, j = 1, 2, \ldots, J$, according to certain criteria, such as minimum Euclidean distance (MED). Specifically, by viewing the mapping function g_j as a $N \times \log_2 M$ complex codebook generator matrix times the jth user's bit vector $\mathbf{b}_j$, we have

$$\mathbf{c}_j = \mathbf{G}_j \mathbf{b}_j, \tag{4}$$

where $\mathbf{G}_j \in \mathbb{C}^{N \times \log_2 M}$ is the codebook generator matrix of the jth user. Therefore, the codebook for user j is $\mathcal{X}_j = \mathbf{V}_j \mathbf{G}_j \mathbf{B}$.

The received signal of user j in downlink channel after the multiplexing can be expressed as

$$\mathbf{r}_j = \operatorname{diag}(\mathbf{h}_j) \sum_{u=1}^{J} \mathbf{V}_u \mathbf{G}_u \mathbf{b}_u + \mathbf{n}_j, \tag{5}$$

where $\mathbf{h}_u = [h_{j,1}, h_{j,2}, \ldots, h_{j,K}]^T \in \mathbb{C}^{K \times 1}$ is the channel coefficient vector between the base station and the jth user, diag$(\cdot)$ denotes the diagonalization of a matrix and $\mathbf{n}_j = [n_{j,1}, n_{j,2}, \ldots, n_{j,K}]^T$ is the complex Gaussian vector with the variance with zero mean and variance N_0, i.e., $n_{j,k} \sim \mathcal{CN}(0, N_0)$.

In the next section, we will design the near optimal generating matrices $\mathbf{G}_j, j = 1, 2, \ldots, J$ to improve the error rate performance with the proposed novel autoencoder.

III. Proposed novel autoencoder

In this section, a novel autoencoder is presented for downlink SCMA systems. We first present the SCMA mapping, i.e., the signal model in (4), inspired encoder designed. Then, the multi-user detection with deep multi-task learning is elaborated. In addition, training procedure and complexity analysis will be discussed.

A. Autoencoder

Autoencoder is a special class of neural networks which is trained to produce an output data that matches with their input data. It is composed of a basic DNN unit formed of multiple repetitive hidden layers. Each hidden layer is an affine mapping followed by a nonlinearlity operator. The output of the lth hidden layer is given by

$$\mathbf{x}_l = \sigma_l \left(\mathbf{W}_l \mathbf{x}_{l-1} + \mathbf{z}_l \right), \tag{6}$$

where $\boldsymbol{W}_l$, $\mathbf{z}_l$, and σ_l denote the weight matrix, bias vector and the activation function for the lth layer, respectively. The encoder first transforms the input vector $\mathbf{x}$ into hidden representation $\mathbf{y}$ through a deterministic mapping $e_{\boldsymbol{\theta}}$, i.e., $\mathbf{y} = e_{\boldsymbol{\theta}}(\mathbf{x})$, where $\boldsymbol{\theta}$ denotes the parameter set with all the weight matrices and bias vectors. The resulting representation $\mathbf{y}$ is then mapped back to reconstruct the input vector, i.e., $\hat{\mathbf{x}} = d_{\boldsymbol{\theta}'}(\mathbf{y})$. The mapping $d_{\boldsymbol{\theta}}$ is called decoder and $\boldsymbol{\theta}'$ is the corresponding parameter set. The DAE is a type of autoencoder that learns to produce original denoised samples from the inputs contaminated by noise. In an DAE, the parameter set θ and $\boldsymbol{\theta}'$ are trained to minimize the reconstruction error [16]

$$\boldsymbol{\theta}^*, \boldsymbol{\theta}'^* = \underset{\boldsymbol{\theta}, \boldsymbol{\theta}'}{\operatorname{argmin}} \, L\left(\mathbf{x}, d_{\boldsymbol{\theta}'}\left(e_{\boldsymbol{\theta}}(\mathbf{x})\right)\right), \tag{7}$$

where L is a loss function, such as the squared error loss $L(\mathbf{x}, \hat{\mathbf{x}}) = \parallel \mathbf{x} - \hat{\mathbf{x}} \parallel^2$. Another commonly used loss function is the cross-entropy loss $L_{\text{CE}}(\mathbf{x}, \hat{\mathbf{x}}) = -\sum_{d=1}^{D} x_d \log(\hat{x}_d)$, where D is the length of the output vector, $x_d \in \mathbf{x}$ and $\hat{x}_d \in \hat{\mathbf{x}}$. Note that for cross-entropy loss, $\mathbf{x}$ and $\hat{\mathbf{x}}$ are in the form of the bit vector and bit probability, respectively.

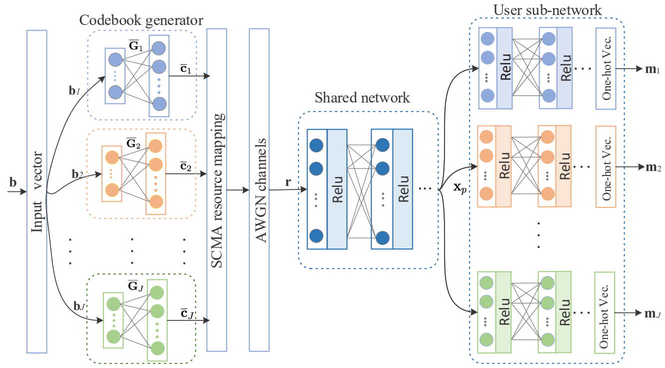

Fig. 1. The system structure of the proposed E2E-SCMA.

B. Signal Model Inspired Encoder Design

In our proposed E2E-SCMA, the mapping from the jth data stream to the jth user's constellation, i.e., $\mathbf{c}_j = g_j(\mathbf{b}_j)$ is implemented with neural networks. Note that the SCMA encoding in (4) has the same expression with neural network in (6) when the activation function is linear with basis $\boldsymbol{z} = \mathbf{0}^T$. Therefore, the codebook generation process, i.e., g_j, can be implemented with a simple neural network, which only consists of the input layer and output layer. The weight matrix in the neural network is equivalent to the generator matrix $\mathbf{G}_j$. Since the proposed network operates in real domain, the output is separated into real and imaginary parts. Hence, (4) is re-written as

$$\bar{\mathbf{c}}_j = \bar{\mathbf{G}}_j \mathbf{b}_j, \tag{8}$$

with

$$\bar{\mathbf{c}}_j = \begin{bmatrix} \Re(\mathbf{c}_j) \\ \Im(\mathbf{c}_j) \end{bmatrix}, \bar{\mathbf{G}}_j = \begin{bmatrix} (\mathbf{G}_j^{\mathrm{R}})^T & (\mathbf{G}_j^{\mathrm{I}})^T \end{bmatrix}^T, \tag{9}$$

where $\mathbf{G}_j^{\mathrm{R}}$ and $\mathbf{G}_j^{\mathrm{I}}$ are the generator matrices of the real and imaginary parts, respectively. Based on the above analysis, the proposed model based E2E-SCMA with J users is shown in Fig. 1, where the proposed E2E-SCMA is composed of J codebook generators, a signature mapping module, a channel module, and a multi-user detection module. The structure of codebook generator is inspired by the signal model and only consists two layers, i.e., the input layer and the output layer. In addition, the number of nodes for input layer and output layer are $\log_2(M)$ and $2N$, respectively.

In the forward-propagation phase, source message vector $\mathbf{b}_j$ first flows through codebook generator network, parameterized by $\bar{\mathbf{G}}_j$ to derive the multi-dimensional complex symbol $\bar{\mathbf{c}}_j$, and then the symbols are mapped to SCMA resources according to $\mathbf{V}_j$. After that, J users' data symbols are superimposed before passing through a Gaussian channel[2]. Finally, the superimposed signal is decoupled to accurately recover source messages based on task-specific sub-networks in the decoder, which will be elaborated in the next subsection.

C. Decoder Design with Multi-task Learning

At the decoder part, deep multi-task learning is adopted to design the multi-user detector. The proposed decoder consists of one shared network and J user specific sub-networks, where the shared network is designed for exchanging the information between the subcarriers and the jth task is responsible for recovering the jth user's data. We employ one-hot vector to represent the input binary message vector $\mathbf{b}_j$, namely, each message $\mathbf{b}_{j,m}, m \in \{1, 2, \dots, M\}$ is represented by an M-dimensional one-hot vector

[2]In this paper, we focus on the Gaussian channel case as in [11], [13], [14] in order to give a clear comparison with other benchmarks. The fading channel will be investigated in future work.

$\mathfrak{m}_j^m$, which is the mth column of the identity matrix $\mathbf{I}_M$. For example, for $M = 4$, the one-hot mapping is defined as

$$\begin{aligned}
\mathbf{b}_{j,1} &= [-1,-1]^T \leftrightarrow \mathfrak{m}_j^1 = [1,0,0,0],\\
\mathbf{b}_{j,2} &= [-1,+1]^T \leftrightarrow \mathfrak{m}_j^2 = [0,1,0,0],\\
\mathbf{b}_{j,3} &= [+1,-1]^T \leftrightarrow \mathfrak{m}_j^3 = [0,0,1,0],\\
\mathbf{b}_{j,4} &= [+1,+1]^T \leftrightarrow \mathfrak{m}_j^4 = [0,0,0,1].
\end{aligned} \tag{10}$$

The decoder can be expressed as $d^{\mathrm{P}}_{\overline{\boldsymbol{\theta}}} d^{\mathrm{U}}_{\boldsymbol{\theta}_j} : \boldsymbol{r}_j \to \boldsymbol{p}_j$, where $d^{\mathrm{P}}_{\overline{\boldsymbol{\theta}}}$ and $d^{\mathrm{U}}_{\boldsymbol{\theta}_j}$ are the non-linear mapping of the forward DNN for the shared network and the jth user' sub-network, respectively. $\boldsymbol{p}_j$ is the output messages, $\overline{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}_j$ are the parameter sets of the shared network and the jth user' sub-network, respectively. In our implementation, we choose fully-connected DNN with L_{P} and L_{U} layers for both shared network and user sub-network. The above process can be expressed as

$$\begin{aligned}
\mathbf{p}_j = d^{\mathrm{U}}_{\boldsymbol{\theta}_j}(\mathbf{x}_{\mathrm{P}}) &= \sigma^{\mathrm{U}}_{j,L_{\mathrm{U}}}\Big(\mathbf{W}_j^{(L_{\mathrm{U}})}\Big(\sigma^{\mathrm{U}}_{j,L_{\mathrm{U}}-1}\cdots\\
&\quad \sigma^{\mathrm{U}}_{j,1}\left(\mathbf{W}_j{}^{(1)}\mathbf{x}_{\mathrm{P}} + \mathbf{z}_j{}^{(1)}\right)\cdots + \mathbf{z}_j^{(L_{\mathrm{U}}-1)}\Big) + \mathbf{z}_j{}^{(L_{\mathrm{U}})}\Big),\\
\mathbf{x}_{\mathrm{P}} = d^{\mathrm{P}}_{\overline{\boldsymbol{\theta}}}(\mathbf{r}) &= \sigma^{\mathrm{P}}_{L_{\mathrm{P}}}\Big(\overline{\mathbf{W}}^{(L_{\mathrm{P}})}\Big(\sigma^{\mathrm{P}}_{L_{\mathrm{P}}-1}\cdots\\
&\quad \sigma^{\mathrm{P}}_{1}\left(\overline{\mathbf{W}}^{(1)}\mathbf{r}_j + \overline{\mathbf{z}}^{(1)}\right)\cdots + \overline{\mathbf{z}}^{(L_{\mathrm{P}}-1)}\Big) + \overline{\mathbf{z}}^{(L_{\mathrm{P}})}\Big),
\end{aligned} \tag{11}$$

where $\mathbf{x}_{\mathrm{P}}$ is the output of the shared layer, σ^{P}_l and $\sigma^{\mathrm{U}}_{j,l}$ denote the activation function of the lth layer of shared network and the jth sub-network, respectively. $\boldsymbol{\theta}_j = \left\{\mathbf{W}_j^{(1)}, \mathbf{z}_j^{(1)}, \ldots, \mathbf{W}_j^{(L_{\mathrm{U}})}, \mathbf{z}_j^{(L_{\mathrm{U}})}\right\}$, and $\overline{\boldsymbol{\theta}} = \left\{\overline{\mathbf{W}}^{(1)}, \overline{\mathbf{z}}^{(1)}, \ldots, \overline{\mathbf{W}}^{(L_{\mathrm{P}})}, \overline{\mathbf{z}}^{(L_{\mathrm{P}})}\right\}$ are the parameters to be learned.

Observing that the task of SCMA detection is to recover the source messages in a limited search space, such a problem is equivalent to a typical classification problem in the machine learning field. Hence, this motivates us to employ the widely used softmax activation for output layer. To facilitate the network convergence, ReLU activation function is adopted for hidden layers. Assume that the input of softmax is a vector $\mathbf{w}_j$ of dimension M, and $w_{j,m}$ is the mth entry of $\mathbf{w}_j$. Then, the softmax activation function takes the following expression:

$$p_{j,m} = \frac{\exp(w_{j,m})}{\sum_{m'=1}^{M}\exp(w_{j,m'})}, \tag{12}$$

where $p_{j,m}$ is the mth entry of the output $\mathbf{p}_j$ with $\sum_{m=1}^{M} p_{j,m} = 1$. All hidden layers adopt ReLU activation function, which can facilitate the network convergence during the training process. As for the loss function, we consider the corresponding softmax cross-entropy loss for each user. Let $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_J^T]^T$ and $\mathfrak{m} = [\mathfrak{m}_1^T, \mathfrak{m}_2^T, \ldots, \mathfrak{m}_J^T]^T$, where $\mathfrak{m}_j$ is the one hot representation of $\mathbf{b}_j$. The overall loss function is the summation over J users, which can be expressed as

$$L^{\text{E2E-SCMA}}(\mathbf{p}, \mathfrak{m}) = -\sum_{j=1}^{J}\sum_{m=1}^{M} q_{j,m}\log(p_{j,m}), \tag{13}$$

where $q_{j,m}$ denotes the mth entry of $\mathfrak{m}_j$. The loss function measures the difference between predicted probability $\mathbf{p}$ diverges from the actual label $\mathfrak{m}$. Therefore, we aim to seek the model parameters $\overline{\mathbf{G}}_j, \overline{\boldsymbol{\theta}}, \boldsymbol{\theta}_j$ to minimize the overall loss:

$$\{\overline{\mathbf{G}}_j{}^*, \overline{\boldsymbol{\theta}}^*, \boldsymbol{\theta}_j^*\} = \underset{[\overline{\mathbf{G}}_j]_{j=1}^{J}, \overline{\boldsymbol{\theta}}, [\boldsymbol{\theta}_j]_{j=1}^{J}}{\arg\min} \; L^{\text{E2E-SCMA}}(\mathbf{p}, \mathfrak{m}). \tag{14}$$

D. Training Algorithm

The encoder and decoder are jointly optimized with gradient decent based method using forward and backward propagation, such as adaptive moment estimation (ADAM). **Algorithm 1** demonstrates the detailed training of the proposed E2E-SCMA system. $\bar{\mathbf{G}}_j, j = 1, 2, \ldots, J$ are first initialized with Huawei

Algorithm 1 Training of E2E-SCMA.

Initialization: Set J, K, $\mathbf{V}_j$, α_0, β, D, $E_b/N_{0_{\min}}$, $E_b/N_{0_{\max}}$, I_T and initialize the network parameters $\overline{\mathbf{G}}_j, \overline{\boldsymbol{\theta}}$, $\boldsymbol{\theta}_j, j \in \{1, 2, \ldots, J\}$.

repeat

$t \leftarrow 1$

Randomly generate training samples $\mathbf{b}_j$ and transfer $\mathbf{b}_j$ to one-hot vector $\mathfrak{m}_j$

Froward Propagation

SNR $\leftarrow \mathcal{U}\left(E_b/N_{0_{\min}}, E_b/N_{0_{\max}}\right)$, $\alpha_t \leftarrow \alpha_0 \beta^{(t/D)}$

$\bar{\mathbf{c}}_j \leftarrow \mathbf{b}_j$ according to (8) and (9)

$\mathbf{r} \leftarrow$ Obtain $\mathbf{r}$ after resource mapping and pass channel

$\boldsymbol{p}_j \leftarrow d^{\mathrm{P}}_{\overline{\boldsymbol{\theta}}} \cdot d^{\mathrm{U}}_{\boldsymbol{\theta}_j}(\boldsymbol{r})$ according to (11)

$L^t{}_{batch} \leftarrow L^{\text{E2E-SCMA}}(\mathbf{p}, \mathfrak{m})$ according to (13)

Backward Propagation

$\overline{\mathbf{G}}_j$, $\boldsymbol{\theta}_j, \overline{\boldsymbol{\theta}} \leftarrow$ Update $\boldsymbol{\theta}_j, \overline{\boldsymbol{\theta}}$ with α_t, $\nabla_{\boldsymbol{\theta}_j, \overline{\boldsymbol{\theta}}} L^t{}_{batch}$, and $\overline{\mathbf{G}}_j$ with α_t and $\nabla_{\overline{\mathbf{G}}_j, \overline{\boldsymbol{\theta}}, \boldsymbol{\theta}_j} L^t{}_{batch}$ with gradient-based optimizer

$t \leftarrow t + 1$

until reaching the maximum iteration number I_T

codebook [17]. Specifically, we first obtain $\mathbf{G}_j = \mathbf{C}_j \mathbf{B}^T (\mathbf{B}\mathbf{B}^T)^{-1}$, where $\mathbf{C}_j$ denotes the jth user's codebook in [17] by removing the zero dimensions. Then, $\bar{\mathbf{G}}_j$ is obtained by concatenating the real and imaginary parts of $\mathbf{G}_j$. The weights of the decoder, i.e., $\bar{\boldsymbol{\theta}}$, and $\boldsymbol{\theta}_j$, are initialized with a normal distribution with mean 0 and variance 1. In the forward propagation, the randomly generated input data first flows through the encoder and decoder to obtain an estimation of the input message. Then, during the backward propagation, the parameters $\overline{\mathbf{G}}_j, \overline{\boldsymbol{\theta}}$, and $\boldsymbol{\theta}_j$ are updated by minimising the total loss. In addition, the learning rate α_t decays exponentially at each iteration t with a decay factor of β and decay step of D. With respect to the training E_b/N_0, the authors in [11], [13]–[15] obtained SCMA codebooks by training the system at a fixed E_b/N_0. However, in our implementation, the training SNR for each iteration was randomly generated so that the SNR will be uniformly distributed on $\mathcal{U}\left(E_b/N_{0_{\min}}, E_b/N_{0_{\max}}\right)$. This approach allows us to train an SCMA system to work over a wide range of SNR values while maintaining a low error rate performance.

E. Complexity Analysis

The main differences between E2E-SCMA and convention SCMA in terms of complexity is the decoder part, i.e., DNN decoder and MPA. Hence, we main focus on analyze the complexity of DNN decoder and MPA. The complexity of MPA is given by $\mathcal{O}\left(N_{iter} K d_f^2 M^{df}\right)$ [9], where N_{itr} is defined as the iteration number of MPA. For E2E-SCMA, we are concerned about the complexity of online deployment. The main computation in E2E-SCMA is matrix multiplication, which is dominated by the two consecutive layers with the largest number of neural nodes. Therefore, we can simply the computation complexity as $\mathcal{O}\left(L_1 L_2\right)$, where L_1 and L_2 are the largest number of neural nodes of two consecutive layers.

IV. Numerical results

In this section, we evaluate the error rate performance of the proposed E2E-SCMA system in Gaussian channel. The following indicating matrix with $J = 6, K = 4, N = 2$ is given by

$$\mathbf{F}_{\mathbf{4}\times\mathbf{6}} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{15}$$

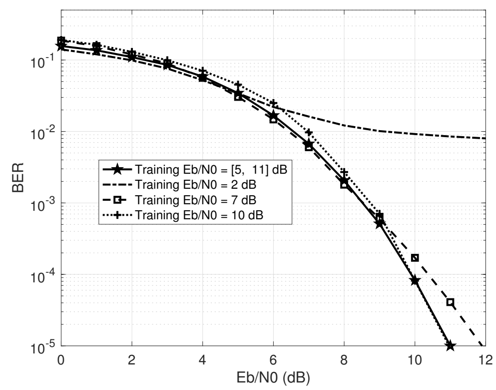

Fig. 2. System performance of E2E-SCMA trained by various E_b/N_0.

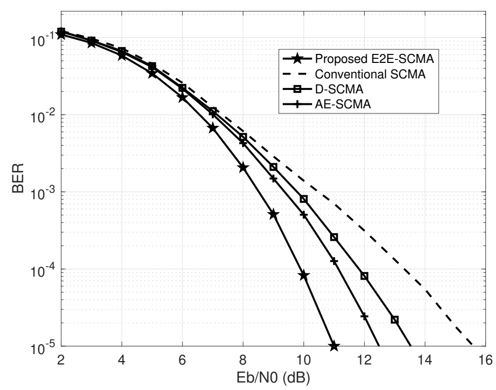

Fig. 3. BER performance comparison with different decoders.

The initial learning rate, decay step and decay factor are set to be $\alpha_0 = 0.001$, $D = 500$ and $\beta = 0.9$, respectively. The batch size for each iteration is set to be 1000 for a trade-off between convergence rate and computational efficiency. The maximum iteration number is $I_T = 2000$. Therefore, the total number of training samples is 2×10^6. We choose a wide range of training E_b/N_0, specifically, we set $E_b/N_{0_{\min}} = 5$ dB and $E_b/N_{0_{\max}} = 11$ dB. The codebook generator is implemented with $\log_2(M)$ input nodes and $2 \times K$ output nodes. For the decoder, the number of nodes and hidden layers for shared network are $\{128, 64\}$ and $L_P = 2$, respectively, whereas the two parameters for user sub-network are $\{64, 32, 16\}$ and $L_U = 3$, respectively. Therefore, the complexity of the E2E-SCMA is $\mathcal{O}\left(L_1 L_2\right)$, where $L_1 = 128$ and $L_2 = 64$.

Since the values of E_b/N_0 in training influence the BER performance, we investigate how training samples generated by different E_b/N_0 can affect the system performance in Fig. 2. We first train the system at the fixed E_b/N_0 values, which were set to be $E_b/N_0 = 2$ dB, 7 dB and 10 dB, respectively. Then, the system was also trained in the E_b/N_0 range $\mathcal{U}\left(5, 11\right)$ dB. It is clearly shown that the low E_b/N_0 trained network only performs well in the low E_b/N_0 range, whereas the high E_b/N_0 trained network will degrade the performance in the low E_b/N_0 range. A better way is to train the network in a wide E_b/N_0 range, thus the trained system can harvest the good performance over a wide range E_b/N_0 .

In Fig. 3, we compare the BER performance of the proposed E2E-SCMA scheme with the AE-SCMA scheme [11], the D-SCMA scheme [13], and the conventional SCMA scheme with Huawei codebook [17]. The MPA decoder is employed for conventional SCMA scheme to compare with deep learning designed SCMA system. The results show that the proposed scheme significantly outperforms all conventional SCMA schemes. Specifically, the proposed E2E-SCMA achieves 3.5 dB gain and 1.8 dB gain over D-SCMA, AE-SCMA scheme at SER $= 10^{-5}$, respectively.

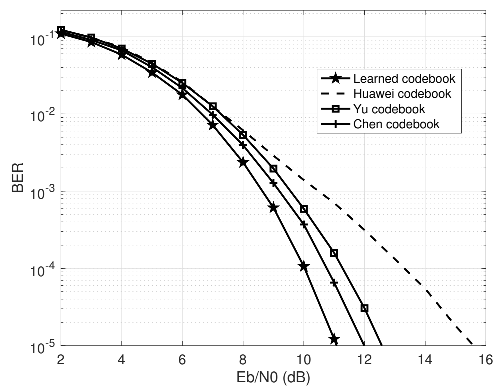

Fig. 4. The BER performance of different codebooks with MPA decoder.

TABLE I
A COMPARISON OF MEDS OF DIFFERENT CODEBOOKS

Codebook	MED
Huawei [17]	0.56
Yu [6]	0.90
Chen [5]	1.07
Learned codebook	1.17

To evaluate the codebook obtained by E2E-SCMA scheme, we compare the MED and corresponding BER performance with MPA decoder with the state of art codebooks. The MED is obtain by calculating $M^J\left(M^J-1\right)/2$ mutual distances between M^J superimposed codewords, which constitute a superimposed constellation Φ. Hence, the MED can be expressed as

$$\min\left\{\| \mathbf{v}_n - \mathbf{v}_m \|^2, \forall \mathbf{v}_n, \mathbf{v}_m \in \Phi, \forall m, n \in Z_{M^J}, m \neq n\right\},$$

where Z_{M^J} stands for the integer set $\left\{1,2,\ldots,M^J\right\}$. Specifically, the MED of learned codebook is compared with Huawei codebook [17], Chen codebook [5] and Yu codebook [6]. The results are presented in Table I. It can be seen that the learned codebook owns MED = 1.17 and is higher than other codebooks. Then, BER comparisons of different codebooks with MPA decoder are shown in Fig. 4. The proposed codebook achieves 4.8 dB gain over the Huawei codebook at BER $=10^{-5}$, about 1.8 dB gain over the Yu codebook, and 1 dB gain over the Chen codebook at BER $=10^{-5}$. The proposed codebook and the codebooks employed for comparison are all available at our GuitHub project[3].

V. CONCLUSION

In this paper, we have proposed an E2E-SCMA by joint optimization of SCMA encoder and decoder with the aid of DAE. Our key idea is to design the SCMA encoder by taking into account of the mapping procedure and then optimize the decoder with multi-task learning approach. Simulation results showed that the use of multi-task learning technique enables efficient derivation of codebook and decoding strategy for a sparse and multidimensional superimposed signal. In addition, our proposed scheme outperforms conventional schemes and existing autoencoder SCMA in terms of both error rate and computational complexity.

REFERENCES

[1] Z. Liu and L.-L. Yang, "Sparse or dense: A comparative study of code-domain NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4768–4780, Aug. 2021.

[2] Q. Luo *et al.*, "An error rate comparison of power domain non-orthogonal multiple access and sparse code multiple access," *IEEE Open J. Commun. Soc.*, vol. 2, no. 4, pp. 500–511, Mar. 2021.

[3]https://github.com/ethanlq/SCMA-codebook/tree/main/CB_autoencoder

[3] H. Nikopour and H. Baligh, “Sparse code multiple access,” in *Proc. IEEE 24th Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, London, U.K., Sep. 2013, pp. 332–336.

[4] R. Hoshyar, F. P. Wathan, and R. Tafazolli, “Novel low-density signature for synchronous cdma systems over awgn channel,” *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1616–1626, Apr. 2008.

[5] Y.-M. Chen and J.-W. Chen, “On the design of near-optimal sparse code multiple access codebooks,” *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2950–2962, May 2020.

[6] L. Yu, P. Fan, D. Cai, and Z. Ma, “Design and analysis of SCMA codebook based on star-QAM signaling constellations,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10 543–10 553, Nov. 2018.

[7] K. Lai *et al.*, “Analyzing uplink grant-free sparse code multiple access system in massive IoT networks,” *IEEE Internet Things J.*, Sep. 2021.

[8] L. Yu *et al.*, “Sparse code multiple access for 6G wireless communication networks: Recent advances and future directions,” *IEEE Commun. Stand. Mag.*, vol. 5, no. 2, pp. 92–99, Apr. 2021.

[9] L. Yang, Y. Liu, and Y. Siu, “Low complexity message passing algorithm for SCMA system,” *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2466–2469, Dec. 2016.

[10] Z. Mheich, L. Wen, P. Xiao, and A. Maaref, “Design of SCMA codebooks based on golden angle modulation,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1501–1509, Feb. 2018.

[11] J. Lin *et al.*, “A novel deep neural network based approach for sparse code multiple access,” *Neurocomputing*, vol. 382, no. 8, pp. 52–63, Mar. 2020.

[12] T. O’shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[13] M. Kim, N.-I. Kim, W. Lee, and D.-H. Cho, “Deep learning-aided SCMA,” *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 720–723, Jun. 2018.

[14] M. Han, H. Seo, A. T. Abebe, and C. G. . Kang, “Deep learning-based multi-user multi-dimensional constellation design in code domain non-orthogonal multiple access,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[15] C.-P. Wei *et al.*, “SCMA decoding via deep learning,” *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 878–881, Apr. 2021.

[16] Y. Zhang and Q. Yang, “A survey on multi-task learning,” 2017, arXiv: 1707.08114. [Online]. Available: https://arxiv.org/abs/1707.08114

[17] Altera Innovate Asia website, Presentation, “1st 5G algorithm innovation competition-env1.0-SCMA.” [Online]. Available: http://www.innovateasia.com/5G/en/gp2.html.