Multi-Task Learning Aided Joint Constellation Design and Multiuser Detection for GF-NOMA

Zhe Ma*, Wen Wu†, Feifei Gao*, and Xuemin (Sherman) Shen†
*Department of Automation, Tsinghua University, Beijing, China
†Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
Email: maz16@mails.tsinghua.edu.cn, w77wu@uwaterloo.ca, feifeigao@ieee.org, sshen@uwaterloo.ca

***Abstract*—This paper aims to investigate the joint optimization of multidimensional constellation design (MCD) and multiuser detection (MUD) for grant-free non-orthogonal multiple access (GF-NOMA). We first formulate the joint optimization problem and derive its explicit expression using variational inference. Due to the intractability of the joint optimization problem, we then resort to deep learning (DL) and approximate the optimal solution in an end-to-end manner. Specifically, we develop a novel variational autoencoder based network, such that the distribution of the multidimensional constellations can be accessed and optimized. We also design a multi-task learning architecture on the decoder side to deal with the complex coupling among signal streams, by taking the MUD process as multiple distinctive yet related tasks. The derivation of the loss function for network training is presented, and simulation results are provided to validate the superior performance of the proposed method over conventional approaches.**

I. INTRODUCTION

Grant-free non-orthogonal multiple access (GF-NOMA) is a promising solution for massive-type communication (mMTC) and the Internet of Things (IoT), since it exploits both the benefits of non-orthogonal signal superposition as well as grantless access mechanism [1]. By non-orthogonally allocating radio resources among devices, GF-NOMA can support massive connectivity with limited resources. Besides, GF-NOMA allows users to autonomously transmit their data without preceding scheduling process, which can greatly reduce the signaling overhead required for the coordination between the base station (BS) and users.

The two most important components in GF-NOMA are multidimensional constellation design (MCD) and multiuser detection (MUD), where MCD assigns uniquely decodable symbols for users while MUD recovers user messages by utilizing the distinctions among these symbols. There have been numerous works investigating the MCD and MUD for GF-NOMA. For MCD, low-density sequences [2], constellation rotation [3], and golden angle modulation [4] were exploited to mitigate inter-user interference. For MUD, the factor graph-based message passing algorithm (MPA) was used in sparse code multiple access (SCMA) [5]. Successive interference cancellation (SIC) was adopted in [6], which distinguishes different users based on the power difference. Moreover, compressed sensing (CS) techniques were introduced in MUD by taking advantage of the sporadic transmission [7].

However, the aforementioned works isolate the design of MCD and MUD, which is suboptimal according to the data processing theorem [8]. A joint optimization approach is thus necessitated to fully unlock the benefit of GF-NOMA. Nonetheless, it is quite challenging to tackle such a joint optimization problem due to the intractable system model caused by the complicated signal superposition [9]. This motivates us to utilize deep learning (DL) techniques that can approximate the optimal solution by tapping the universal approximation theorem of deep neural networks (DNNs).

Thanks to the strong ability to solve intricate and intractable problems, DL has been widely applied in wireless communications [10] [11]. One of the most promising applications is end-to-end communication, as it provides an effective paradigm to jointly optimize all communication blocks. The key idea of end-to-end communication is to interpret the whole communication system as an autoencoder, where both transmitter and receiver are implemented as neural networks and optimized in an end-to-end manner. The idea was first pioneered in [12], and has led to many extensions [13] [14]. However, existing works on end-to-end communication are usually based on the traditional autoencoder, where the encoder outputs a single value that may be insufficient to describe the constellation attribute. On the contrary, variational autoencoder provides a probabilistic manner by outputting the mean value and variance of the encoded vectors, through which we can make insight into the constellation distribution [15]. Moreover, the signal streams in GF-NOMA are intricately superposed, and the detection objectives of different users are mutually conflicting, which brings about challenges for traditional autoencoders.

In this paper, we propose a DL-based method to jointly optimize MCD and MUD for GF-NOMA. Specifically, we develop a variational autoencoder network and incorporate it with multi-task learning to tackle the complex signal coupling. Simulation results reveal that the proposed method enables significant gains compared to conventional methods.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

As illustrated in Fig. 1, we consider an uplink GF-NOMA system with one BS and N users, where all the terminals are equipped with a signal antenna. We assume that the nth user wishes to send a $\log_2 M$-bits message $s_n \in \{1, \cdots, M\}$ to the BS. The message s_n is modulated into a K-dimensional symbol $\mathbf{x}_n = [x_{n,1}, \cdots, x_{n,K}]^T \in \mathbb{C}^{K\times 1}$ with a power constraint $||\mathbf{x}_n|| \le K$. Particularly, we consider the typical overloaded

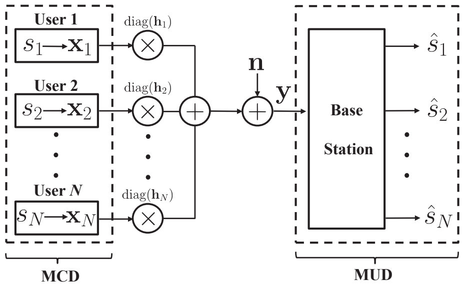


Fig. 1. System model of the uplink GF-NOMA system.

GF-NOMA system, where the number of users is larger than the symbols' dimension, i.e., $N > K$. Therefore, signals from all users are superimposed and transmitted over K orthogonal resources (e.g., time slots or OFDM subcarriers). The received signal at the BS can be represented as

$$\mathbf{y} = \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n)\mathbf{x}_n + \mathbf{n}, \tag{1}$$

where $\mathbf{h}_n = [h_{n,1}, \cdots, h_{n,K}]^T \in \mathbb{C}^{K\times 1}$ is the channel vector from the nth user to the BS over K resources, and $\mathbf{n}$ is the noise vector that follows the complex Gaussian distribution $\mathcal{CN}(0, \sigma^2\mathbf{I}_K)$ with $\mathbf{I}_K$ being a $K \times K$ identity matrix.

Upon receiving the superimposed signal $\mathbf{y}$, the BS tries to retrieve all the user messages through MCD process. The overall communication procedure of user n can be mathematically viewed as the cascade of the MCD process and the MUD process, which is given by

$$\hat{s}_n = g_n\left(\sum_{n=1}^{N} \text{diag}(\mathbf{h}_n) f_n(s_n) + \mathbf{n}\right), \tag{2}$$

where $\hat{s}_n$ is the estimate of the original message s_n, $f_n : s_n \rightarrow \mathbf{x}_n$ is the user-specific MCD function that maps s_n into $\mathbf{x}_n$, and $g_n : \mathbf{y} \rightarrow \hat{s}_n$ is the MUD function that recovers s_n from $\mathbf{y}$.

B. Problem Formulation

The main purpose of the joint optimization is to find the optimal MCD/MUD pair $([f_n^*]_{n=1}^N, [g_n^*]_{n=1}^N)$ that minimizes the error probability $\sum_{n=1}^N \mathbb{P}(\hat{s}_n \neq s_n)$. We start with the optimal MUD, which can be obtained by maximizing the following marginal likelihood

$$g_n^* = \arg\max_{g_n} P(s_n; g_n) = \arg\max_{g_n} \int P(s_n|\mathbf{y}; g_n) P(\mathbf{y})\, d\mathbf{y}, \tag{3}$$

where $P(\mathbf{y})$ is the probability distribution of $\mathbf{y}$ and $P(s_n|\mathbf{y}; g_n)$ is the posterior probability that g_n correctly recovers s_n given $\mathbf{y}$.

Taking all the N signal streams into account, we can obtain the optimal MUD as

$$[g_n^*]_{n=1}^N = \arg\max_{g_n, \forall n \in \mathfrak{N}} \prod_{n=1}^{N} P(s_n; g_n), \tag{4}$$

where $\mathfrak{N} = \{1, 2, \cdots, N\}$. Since message s_n is independent with each other, we can further simplify (4) as

$$[g_n^*]_{n=1}^N = \arg\max_{g_n, \forall n \in \mathfrak{N}} P(\mathbf{s}; [g_n]_{n=1}^N) = \arg\max_{g_n, \forall n \in \mathfrak{N}} \log P(\mathbf{s}; [g_n]_{n=1}^N), \tag{5}$$

where $\mathbf{s} = [s_1, \cdots, s_N]$ and $P(\mathbf{s}; [g_n]_{n=1}^N) = \int P(\mathbf{s}|\mathbf{y}; [g_n]_{n=1}^N) P(\mathbf{y})\, d\mathbf{y}$. However, it is non-trivial to directly optimize $\log P(\mathbf{s}; [g_n]_{n=1}^N)$ over $[g_n]_{n=1}^N$, since the multidimensional integral $P(\mathbf{s}; [g_n]_{n=1}^N)$ is intractable due to the unknown $P(\mathbf{y})$. According to the law of total probability, $P(\mathbf{y})$ can be presented as $P(\mathbf{y}) = \int P(\mathbf{y}|\mathbf{s}) P(\mathbf{s})\, d\mathbf{s}$, which implies that the MCD mapping should be determined before obtaining $[g_n^*]_{n=1}^N$. Therefore, we resort to the variational inference and rewrite $\log P(\mathbf{s}; [g_n]_{n=1}^N)$ as

$$\begin{aligned}
&\log P(\mathbf{s}; [g_n]_{n=1}^N) \\
&= \int Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) \log P(\mathbf{s}; [g_n]_{n=1}^N)\, d\mathbf{y} \\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[-\log Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) + \log P(\mathbf{y}, \mathbf{s}; [g_n]_{n=1}^N)] \\
&+ KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y}|\mathbf{s}; [g_n]_{n=1}^N)),
\end{aligned} \tag{6}$$

where $Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)$ is an arbitrary parameterized (by $[f_n]_{n=1}^N$) conditional probability distribution of $\mathbf{y}$ given $\mathbf{s}$, and $KL(p(z)||q(z)) = \int p(z) \log \frac{p(z)}{q(z)}\, dz$ is the Kullback-Leibler divergence (KLD) that measures the similarity between two distributions. A more detailed derivation can be found in [15] and is omitted here for brevity.

Since the KLD is non-negative, (6) can be written as

$$\begin{aligned}
&\log P(\mathbf{s}; [g_n]_{n=1}^N) \geq \mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N) \\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[-\log Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) + \log P(\mathbf{y}, \mathbf{s}; [g_n]_{n=1}^N)]
\end{aligned} \tag{7}$$

where $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ is called the *variational lower bound* [16]. Therefore, maximizing $\log P(\mathbf{s}; [g_n]_{n=1}^N)$ is equivalent to maximizing $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$, which can also be written as

$$\begin{aligned}
&\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N) \\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[\log P(\mathbf{s}|\mathbf{y}; [g_n]_{n=1}^N)] \\
&- KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y})) \\
&= \sum_{n=1}^{N} \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[\log P(s_n|\mathbf{y}; g_n)] \\
&- KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y})).
\end{aligned} \tag{8}$$

It is noted that $Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)$ can be regarded as the MCD process that maps s_n to $\mathbf{x}_n, n \in \mathfrak{N}$ and then generates $\mathbf{y}$, as there exists a deterministic mapping between $\mathbf{y}$ and $\mathbf{x}$ in (1). Likewise, $P(s_n|\mathbf{y}; g_n)$ can be considered as the MUD process, which extracts the estimation of s_n from $\mathbf{y}$. Therefore, we readily see that (5) has been converted into a joint optimization

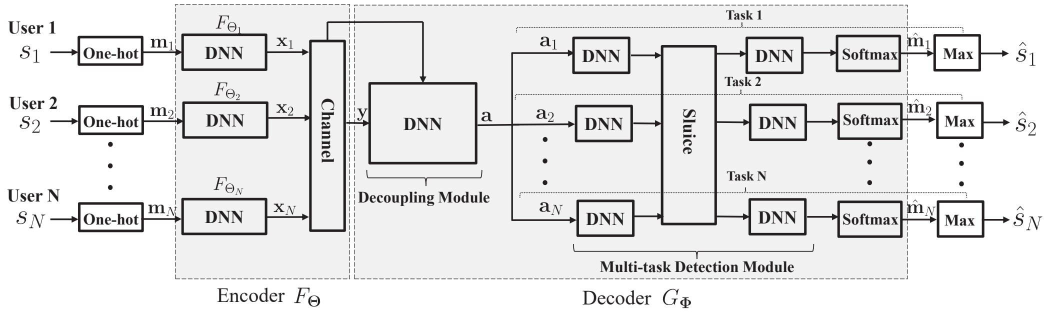


Fig. 2. Architecture of the proposed network.

problem of the MCD and MUD, i.e.,

$$\mathcal{P}1: \underset{f_n, g_n, \forall n \in \mathfrak{N}}{\arg\min} \; -\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N),$$
$$s.t. \; \mathbf{y} = \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n)\mathbf{x}_n + \mathbf{n}. \tag{9}$$

However, it is intractable to solve (9) analytically, due to the infinite searching spaces of f_n and g_n as well as the mutually conflicting objectives of g_n. For this reason, we develop a DL-based method to approximate the optimal solution. Our key idea is to parameterize (9) as a variational autoencoder, where the encoder and the decoder are trained to mimic the optimal MCD and MUD, respectively. The details are elaborated in the next section.

III. DL-Based Joint Optimization for GF-NOMA

A. Proposed Network Architecture

As depicted in Fig. 2, the proposed variational autoencoder network consists of an encoder denoted by $F_{\mathbf{\Theta}}(\cdot)$ and a decoder denoted by $G_{\mathbf{\Phi}}(\cdot)$, where $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are the network parameter sets corresponding to $[f_n]_{n=1}^N$ and $[g_n]_{n=1}^N$, respectively. In essence, the encoder approximates the optimal MCD by learning the optimal mapping between user messages $\mathbf{s}$ and multidimensional symbols $\mathbf{x}_n, n \in \mathfrak{N}$, while the decoder imitates the optimal MUD by learning the optimal mapping from $\mathbf{y}$ back to $\mathbf{s}$. The details and functionalities of the encoder and the decoder are presented as follows, based on which we give the explicit mathematical relations between $\mathbf{\Theta}$ and $[f_n]_{n=1}^N$ as well as $\mathbf{\Phi}$ and $[g_n]_{n=1}^N$.

1) Encoder: As the input of the encoder, we apply the one-hot encoding to represent user messages, which can simplify (9) into a classification problem. In one-hot encoding, each message $s_n \in \{1, \cdots, M\}$ is represented by a M-dimensional vector $\mathbf{m}_n \in \{0,1\}^M$, where the s_nth element is 1 and the others are all 0.

Since it is difficult for a user to exchange information with other users in the GF-NOMA system, the MCD should be performed in a distributed manner. Accordingly, we employ N isolated DNNs at the encoder, denoted as $F_{\mathbf{\Theta}_n}: \mathbf{m}_n \to \mathbf{x}_n, n \in \mathfrak{N}$, where $\mathbf{\Theta}_n$ is the parameter set of the nth DNN. For layer l of $F_{\mathbf{\Theta}_n}$ with N_{l-1} input and N_l output, the output vector can be expressed as

$$\mathbf{z}_n^l = \mathfrak{f}_n^l(\boldsymbol{W}_n^l \mathbf{z}_n^{l-1} + \boldsymbol{b}_n^l), \tag{10}$$

where $\boldsymbol{W}_n^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $\boldsymbol{b}_n^l \in \mathbb{R}^{N_l}$ are the weight matrix and bias, respectively. The hyperbolic tangent (Tanh) function (i.e., $\mathfrak{f}_n^l(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$) is adopted as the activation function in all the hidden layers and output layer. As a result, the overall input-output affine of $F_{\mathbf{\Theta}_n}$ is

$$F_{\mathbf{\Theta}_n}(\mathbf{m}_n) = \mathfrak{f}_n^{L-1}(\mathfrak{f}_n^{L-2}(\cdots(\mathfrak{f}_n^1(\mathbf{m}_n; \mathbf{\Theta}_n^1), \cdots)\mathbf{\Theta}_n^{L-2}); \mathbf{\Theta}_n^{L-1}), \tag{11}$$

where L is the network depth and $\mathbf{\Theta}_n^l = \{\boldsymbol{W}_n^l, \boldsymbol{b}_n^l\}$. A normalization operation is employed at the output layer to satisfy the power constraint. It is worth noting that the communication signals are usually modeled as complex numbers, while most DNNs are based on real-valued operations. Hence, to facilitate the learning process, the complex signals are converted to their real signal version by concatenating their real and imaginary parts. Therefore, the output of the nth DNN is a real-valued vector with $2K$ dimension, and the corresponding constellation is determined as[1]

$$\mathbf{x}_n = F_{\mathbf{\Theta}_n}(\mathbf{m}_n)_{(1:K)} + \sqrt{-1}F_{\mathbf{\Theta}_n}(\mathbf{m}_n)_{(K+1:2K)}, \tag{12}$$

where $F_{\mathbf{\Theta}_n}(\mathbf{m}_n)_{(i:j)}$ is the vector composed of the ith element to the jth element of $F_{\mathbf{\Theta}_n}(\mathbf{m}_n)$. The outputs of all the DNNs are superimposed and then sent over the channel. Substituting (11) into (1), the received signal can be rewritten as

$$\mathbf{y} = \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n)F_{\mathbf{\Theta}_n}(\mathbf{m}_n) + \mathbf{n}. \tag{13}$$

Consequently, we can approximate the MCD process in (9) by [15]

$$Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) = \mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K}), \tag{14}$$

where $\boldsymbol{\mathcal{M}} = [\mathbf{m}_1, \cdots, \mathbf{m}_N]$, $F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) = \sum_{n=1}^N \text{diag}(\mathbf{h}_n)F_{\mathbf{\Theta}_n}(\mathbf{m}_n)$, and $\mathbf{I}_{2K}$ is a $2K \times 2K$ identity matrix. Note that (14) involves the mean value and variance of $\mathbf{y}$, which can be obtained by treating the channel as the output layer of the encoder [15].

[1]Hereafter, for convenience, we assume that all the complex vectors have been converted to the real signal version without changing their mathematical expressions.

2) Decoder: We assume that the channel state information (CSI) is perfectly known at the decoder, since it can be obtained by the BS through pilot-based channel estimation method. As shown in Fig. 2, the decoder mainly comprises two parts, namely the decoupling module and the multi-task detection module. The decoupling module aims to decompress $\mathbf{y}$ to a $2KN$ dimensional vector, and the multi-task detection module incorporates multi-task learning by taking the message recovery of each user as one *task*. Note that without the decoupling module, the multi-task detection module has to recover all user messages directly from $\mathbf{y}$. This wastes the capability of multi-task learning to extract the inherent correlation among different tasks and may degrade the detection performance. Specifically, the decoupled vector can be expressed as

$$\mathbf{a} = G_{\mathbf{\Phi}_D}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N), \tag{15}$$

where $\mathbf{\Phi}_D$ is the parameter set of the decoupling module.

The decoupled vector is evenly split into N vectors $\mathbf{a}_n, n \in \mathfrak{N}$, which are then forwarded to the multi-task detection module. The multi-task detection module can simultaneously optimize all the mutually conflicting detection processes, by taking advantage of the similarities between different tasks [17]. Particularly, we adopt the sluice network that can be trained to mediate the interaction between different tasks, so that the multi-task detection module avoids the costly searching process for potentially optimal relational parameters. The details of the sluice network can be found in [18] and are omitted here for brevity.

It should be mentioned that the last layer of the multi-task detection module is a softmax layer, which ensures that the output of the decoder forms a probability vector $\hat{\boldsymbol{\mathcal{M}}} = [\hat{\mathbf{m}}_1, \cdots, \hat{\mathbf{m}}_N]$ with $||\hat{\mathbf{m}}_n||_1 = 1, n \in \mathfrak{N}$. Accordingly, the MUD process can be approximated as a categorical distribution

$$P(s_n|\mathbf{y}; g_n) = \prod_{i=1}^{M} \hat{\mathbf{m}}_{n_i}^{\mathbf{m}_{n_i}} = \prod_{i=1}^{M} G_{\mathbf{\Phi}}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N)_{n_i}^{\mathbf{m}_{n_i}}, \tag{16}$$

where $G_{\mathbf{\Phi}}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N)_{n_i} = G_{\mathbf{\Phi}_M}(G_{\mathbf{\Phi}_D}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N))_{n_i}$, $\mathbf{\Phi}_M$ is parameter set of the multi-task detection module, and $G_{\mathbf{\Phi}_M}(\cdot)_{n_i}$ is the n_ith element of the multi-detection module output vector.

B. Loss Function

The loss function is a measure of how accurately the neural network is able to predict the expected outcome. When training the neural network, we aim to minimize the loss function by adjusting network parameters. Therefore, to ensure that $\mathbf{\Theta}$ and $\mathbf{\Phi}$ can be fine-tuned to approximate $[f_n^*]_{n=1}^N$ and $[g_n^*]_{n=1}^N$, the loss function should be identical to the objective function of $\mathcal{P}_1$. Substituting (14) and (16) into (9), the loss function can be derived as

$$\begin{aligned}
&\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = \\
&-\sum_{n=1}^{N} \mathbb{E}_{\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})}[\sum_{i=1}^{M} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N)_{n_i})] \\
&+ KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})||P(\mathbf{y})) \\
&= \mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi}) + \mathcal{L}_{KL}(\mathbf{\Theta}),
\end{aligned} \tag{17}$$

where the first right-hand side (RHS) term $\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi}) = -\sum_{n=1}^{N} \mathbb{E}_{\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})}[\sum_{i=1}^{M} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(\mathbf{y}, [\mathbf{h}_n]_{n=1}^N)_{n_i})]$ represents the expected reconstruction loss and the second RHS term $\mathcal{L}_{KL}(\mathbf{\Theta}) = KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})||P(\mathbf{y}))$ is the regularization term that makes $\mathbf{y}$ to follow a prior distribution.

However, computing $\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi})$ involves a sampling process, which stunts the backpropagation since the sampling operation is non-differentiable. To tackle this issue, we leverage the reparameterization trick in [15], whose core idea is to randomly sample ϵ from a unit Gaussian, and then shift the randomly sampled ϵ by the mean value of $\mathbf{y}$ and scale it by the variance of $\mathbf{y}$. Since $\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}[f(z)] = \mathbb{E}_{\mathcal{N}(\epsilon;0,1)}[f(\mu + \sigma\epsilon)] = \frac{1}{S}\sum_{s=1}^{S} f(\mu + \sigma\epsilon^{(s)})$, $\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi})$ can be represented by a differentiable estimator as [15]

$$\begin{aligned}
&\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi}) \\
&= -\sum_{n=1}^{N}\sum_{s=1}^{S}\sum_{i=1}^{M} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) + \boldsymbol{\epsilon}^{(s)}, [\mathbf{h}_n]_{n=1}^N)_{n_i}) \\
&= -\sum_{n=1}^{N} \mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}),
\end{aligned} \tag{18}$$

where $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) = -\sum_{s=1}^{S}\sum_{i=1}^{M} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) + \boldsymbol{\epsilon}^{(s)}, [\mathbf{h}_n]_{n=1}^N)_{n_i})$, S is the number of samples, and $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(0, \frac{\sigma^2}{2}\mathbf{I}_{2K})$.

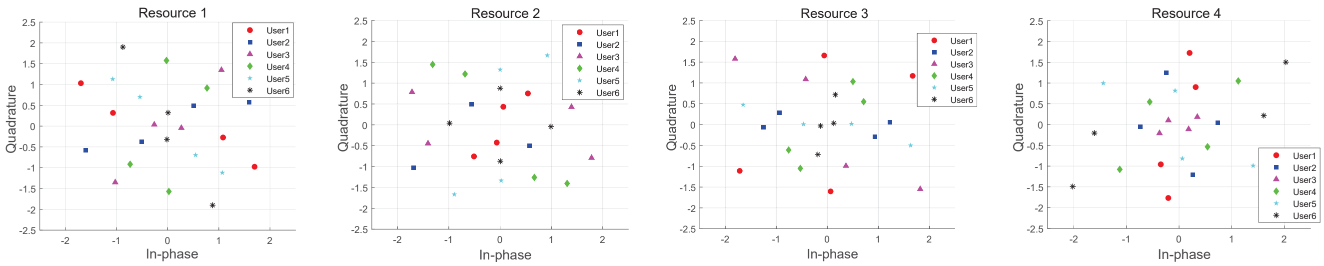
Now the onus remains in determining $P(\mathbf{y})$, with which we can analytically derive $\mathcal{L}_{KL}(\mathbf{\Theta})$ that affects the distribution of learned constellations. According to Shannon's theorem, we know that the transmission rate can be improved by inducing Gaussianity on $\mathbf{y}$. Therefore, we set $P(\mathbf{y}) = \mathcal{N}(P_R\mathbf{I}_{2K}, \frac{\sigma^2}{2}\mathbf{I}_{2K})$ and derive $\mathcal{L}_{KL}(\mathbf{\Theta})$ as

$$\begin{aligned}
&\mathcal{L}_{KL}(\mathbf{\Theta}) \\
&= KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})||\mathcal{N}(P_R\mathbf{I}_{2K}, \frac{\sigma^2}{2}\mathbf{I}_{2K})) \\
&= \frac{2}{\sigma^2}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - P_R\mathbf{I}_{2K})^T(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - P_R\mathbf{I}_{2K}),
\end{aligned} \tag{19}$$

where P_R is the average received signal power.

Combining (17), (18), and (19), the overall loss $\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi})$ can be determined as

$$\begin{aligned}
\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) &= \sum_{n=1}^{N} \mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) \\
&+ \frac{2}{\sigma^2}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - P_R\mathbf{I}_{2K})^T(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - P_R\mathbf{I}_{2K}).
\end{aligned} \tag{20}$$



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

Fig. 3. The projection of the learned multidimensional constellations over 4 orthogonal resources.

It is found in [15] that when batch size is large enough (e.g., more than 100), the sample number S can be set to 1. Meanwhile, if the distribution of user messages, i.e., $P(\boldsymbol{\mathcal{M}})$, is fixed, $F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}})$ is approximately equivalent to $P_R\mathbf{I}_{2K}$ according to the law of large numbers. Therefore, (20) can be expressed as $\mathcal{L}(\mathbf{\Theta},\mathbf{\Phi}) = -\sum_{n=1}^{N}\sum_{i=1}^{M}\mathbf{m}_{n_i}\log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}})+\boldsymbol{\epsilon},[\mathbf{h}_n]_{n=1}^{N})_{n_i})$, which is exactly the cross-entropy loss function that is widely used in DL-based applications.

Remark 1. *Note that when $P(\boldsymbol{\mathcal{M}})$ is variable (e.g., probabilistic encoding) or $\mathbf{n}$ does not follow a zero-mean Gaussian distribution (e.g., interference-limited communication), (20) is different to the cross-entropy loss, which implies that the cross-entropy loss function is no longer optimal. Therefore, classic loss functions may not be suitable in some wireless communication scenarios, which indicates that more insights should be offered into the loss function design. As this problem would merit an independent study and is beyond the scope of this paper, we will investigate it in future work.*

IV. Simulation Results and Evalution

A. Simulation Setup

Our simulation setup is based on the typical NOMA scenario [2], where $N=6$ users share $K=4$ orthogonal resources. The message length is set to 2 bits, i.e., $M=4$, for all users.[2] The encoder of the proposed network consists of 6 DNNs. For each DNN, there are 4 hidden layers with 32 neurons. The decoupling module has 1 DNN comprising 5 hidden layers with 64 neurons. The multi-task detection module is a one-layer sluice network consisting of 12 DNNs and 1 sluice, where each DNN has 5 hidden layers with 64 neurons. The sluice is composed of 1 DNN with 3 hidden layers and 64 neurons for each hidden layer.

The proposed network is Xavier initialized and trained with $80,000$ training datapoints that are randomly sampled from one-hot vectors. These training datapoints are applied as both the input data and the output label, which can be realized by initializing the pseudorandom number generators in users and BS with the same seed. The network is trained at a specific signal to noise ratio (SNR) of 6dB and is tested at a wide range of SNR. The training process is conducted for $5,000$ epochs using stochastic gradient (SGD) algorithm with Adam optimizer. The learning rate is set to be 2×10^{-5} and the batch size is 500.

For comparison, we examine the performance of SCMA with MPA detector (SCMA-MPA), multiuser shared access with MMSE-SIC detector (MUSA-SIC), and the conventional end-to-end network (E2E) [12]. In SCMA-MPA, we apply the user-to-resource mapping and codebook in [19], and set the iteration number of MPA to 5. In MUSA-SIC, the QPSK modulation is adopted and the complex spreading code is generated based on a 3-ary set $\{1,0,-1\}$. In conventional E2E network, the DNNs in both the encoder and the decoder have the same width and depth as those in the proposed network.

B. Simulation Results

We first evaluate the multidimensional constellations learned by the proposed network. For clarity, we project the multidimensional constellations onto a set of orthogonal 2-dimensional signal spaces (i.e., the I and Q channels), where each set of the signal spaces refer to one orthogonal resource. Fig. 3 illustrates the projection of the multidimensional constellations over 4 orthogonal resources, where different colors and shapes represent different users. It can be observed that the learned multidimensional constellations are spread without overlapping in every resource. Specifically, the encoder learns to transmit different messages with various power, which introduces power diversity among users. Besides, the constellations of different users are rotated with different angles around the origin, which further reduces the inter-user interference. These observations indicate that the proposed network can learn to multiplex users by exploiting both the power and phase domains. Furthermore, unlike SCMA where symbols are spanned sparsely, the proposed network allows each user to access all the resources, which may enhance the robustness of transmitting symbols and improve the detection accuracy.

We then compare the average block error rate (BLER) performance, i.e., $\frac{1}{N}\sum_{n=1}^{N}\mathbb{P}(\hat{s}_n\neq s_n)$, of the proposed network with competing methods. Fig. 4 depicts the average BLERs of the proposed network and competing methods under AWGN channel. Notably, DL-based methods (proposed network and E2E) outperform the conventional methods (SCMA-MPA and

[2]It should be noted that the proposed network can be extended to longer messages and larger networks with slight modifications of the network structure.

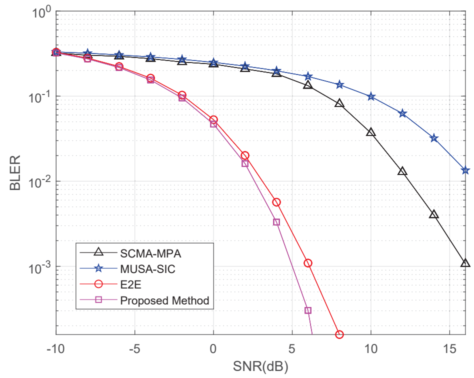


Fig. 4. BLER performance comparisons under the AWGN channel.

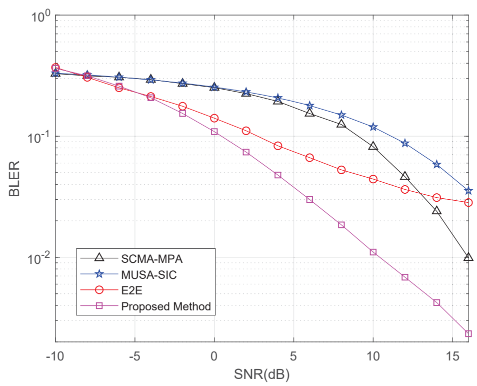


Fig. 5. BLER performance comparisons under the Rayleigh channel.

MUSA-SIC) by a large margin. For example, DL-based methods achieve around 9 dB gain over conventional methods when SNR is higher than 0 dB, which demonstrates the superiority of joint optimization and the potential of end-to-end learning. Moreover, the proposed method is only slightly better than the E2E method, since the latter is sufficient to extract user messages from the simple signal structure in AWGN channel.

Fig. 5 compares the average BLER of the proposed network with competing methods under Rayleigh fading channel. We observe that the proposed network still achieves the lowest BLER among all methods, but the performance gaps between DL-based and conventional methods are narrowed. This is because the transmitting symbols are greatly distorted by fading channels, and it is not easy for DNNs to suppress random fading effects through limited training samples and network size. Besides, the average BLER of the proposed network keeps decreasing as SNR increases, while there are error floors in the average BLER of the E2E method. The reasons are as follows: (i) The optimal detection mappings at different SNRs are hardly related, since the Rayleigh fading channel is randomly generated. The networks are trained at a specific SNR, so it is challenging for them to learn a detection mapping that is optimal at all SNRs. (ii) The proposed network can learn the relationship among different detection tasks through the multi-task learning structure, which is relatively stable compared with random fadings. Hence, the proposed network can learn a good detection mapping, if not the best, for a wide range of SNRs.

V. Conclusion

This paper has proposed a novel DL-based method for the joint design of MCD and MUD in the GF-NOMA system. A variational autoencoder is constructed to approximate the optimal MCD and MUD through training, and multi-task learning is adopted to handle the intricate signal structures. Moreover, the corresponding loss function is derived and analyzed, which may shed light on the design of communication domain knowledge-integrated loss function. Simulation results have demonstrated the effectiveness of the proposed method in terms of BLER performance. We will consider a more practical scenario for our future work, where the users are heterogeneous and have different transmitting rates.

References

[1] M. B. Shahab *et al.*, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1805–1838, May 2020.

[2] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Proc. IEEE PIMRC*, 2013, pp. 332–336.

[3] Y. Zhou *et al.*, "SCMA codebook design based on constellation rotation," in *Proc. IEEE ICC*, 2017, pp. 1–6.

[4] Z. Mheich *et al.*, "Design of SCMA codebooks based on golden angle modulation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1501–1509, Feb. 2019.

[5] J. Dai *et al.*, "Improved message passing algorithms for sparse code multiple access," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9986–9999, Nov. 2017.

[6] L. Yuan *et al.*, "Successive interference cancellation for LDPC coded non-orthogonal multiple access systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5460–5464, Jun. 2018.

[7] Y. Du *et al.*, "Efficient multi-user detection for uplink grant-free NOMA: Prior-information aided adaptive compressive sensing perspective," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2812–2828, Dec. 2017.

[8] J. Ziv and M. Zakai, "On functionals satisfying a data-processing theorem," *IEEE Trans. Inf. Theory*, vol. 19, no. 3, pp. 275–283, May 1973.

[9] L. Bariah, S. Muhaidat, and A. Al-Dweik, "Error performance of NOMA-based cognitive radio networks with partial relay selection and interference power constraints," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 765–777, Jun. 2020.

[10] W. Wu *et al.*, "Fast mmWave beam alignment via correlated bandit learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5894–5908, Dec. 2019.

[11] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, Jan. 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] S. Dorner *et al.*, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[14] B. Zhu *et al.*, "Joint transceiver optimization for wireless communication PHY using neural network," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1364–1373, Jun. 2019.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014, pp. 1–14.

[16] X. Yang, *Understanding the Variational Lower Bound*. 2017, pp. 1–4.

[17] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint*, arXiv:1706.05098, 2017.

[18] S. Ruder *et al.*, "Latent multi-task architecture learning," in *Proc. AAAI*, 2019.

[19] *Altera Innovate Asia 1st 5g algorithm innovation competition.* [Online]. Available: http://www.innovateasia.com