Joint Constellation Design and Multiuser Detection for Grant-Free NOMA

Zhe Ma, *Student Member, IEEE*, Wen Wu, *Member, IEEE*, Mengnan Jian, Feifei Gao, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*— As a promising solution for massive machine-type communication, grant-free non-orthogonal multiple access (GF-NOMA) has received considerable attention in recent years. However, the multidimensional constellation design (MCD) and multiuser detection (MUD) in GF-NOMA are usually optimized in a *divide and conquer* way, leading to local optima and performance degradation. To address this issue, we investigate the joint optimization of MCD and MUD for GF-NOMA. The formulated joint optimization is based on variational inference, which is intractable due to the signal superimposition that makes the optimization variables intricately coupled. Then, we resort to end-to-end deep learning (DL) to obtain the optimal solution. Specifically, we propose a DL-based multi-task variational autoencoder (Mul-VAE) that adopts a variational autoencoder network to optimize the distribution of the constellation points. We further derive the loss function of the proposed network and analyze it from an information-theoretic perspective. On this basis, multi-task learning is employed to deal with mutually conflicting yet related detection processes. Besides, taking heterogeneous transmission rates of users into account, a multi-task prioritizing strategy is designed to balance training performance. Simulation results reveal that the proposed method enables significant gains compared to state-of-the-art techniques.**

***Index Terms*— NOMA, deep learning, mMTC, constellation design, multiuser detection, multi-task learning.**

I. Introduction

GRANT-FREE non-orthogonal multiple access (GF-NOMA) is a favorable paradigm for massive machine-type communication (mMTC) and Internet-of-Things (IoT), as it unlocks the benefits of non-orthogonal signal superposition as well as grant-free access mechanism [1]– [7]. By non-orthogonally allocating radio resources among users, GF-NOMA can support massive connectivity with limited resources. Besides, GF-NOMA allows users to transmit their data without preceding scheduling process, and thus eliminates the signaling overhead required for the coordination between the base station (BS) and users. In general, GF-NOMA consists of two essential components: multidimensional constellation design (MCD) and multiuser detection (MUD), where MCD assigns uniquely decodable symbols for users while MUD recovers user messages by exploiting the distinctions among these symbols.

For MCD, different users were assigned with different power levels according to their channel quality to exploit power diversity [8]. In [9]– [10], low-density sequences and low cross-correlation sequences were utilized to mitigate inter-user interference. In [11], the authors designed multidimensional constellation by maximizing the minimum Euclidean distance. Besides, some other methodologies were also studied in the literature, such as constellation-constrained capacity maximization [12], constellation rotation [13], constellation segmentation [14], and golden angle modulation [15]. For MUD, the factor graph based message passing algorithm (MPA) was used in sparse code multiple access (SCMA) [16]. In [17], successive interference cancellation (SIC) was utilized, which distinguishes different users based on the transmit power. Moreover, compressed sensing (CS) was introduced in MUD to take advantage of the sporadic transmission characteristic [18]– [19].

However, the aforementioned works separately design of MCD and MUD, which are suboptimal according to the data processing theorem [20]. Hence, a joint optimization approach is necessitated to push the performance of GF-NOMA towards the boundary. Nonetheless, it is quite challenging to tackle the joint optimization problem, since the signal superimposition in GF-NOMA incurs complicated coupling relation among optimization variables [21]. To address this issue, we employ deep learning (DL) techniques that are capable of approximating the optimal solution according to the universal approximation theorem of deep neural networks (DNNs) [22].

Thanks to the strong capability to solve intricate and intractable problems, DL has been widely applied in wireless communications [23]– [33], such as signal detection [27], channel estimation [28], and constellation design [29]. Besides enhancing individual communication blocks, DL is also capable of jointly optimizing all components using the concept of

Manuscript received December 16, 2020; revised April 8, 2021 and July 13, 2021; accepted August 25, 2021. Date of publication September 6, 2021; date of current version March 10, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102401 and in part by the National Natural Science Foundation of China under Grant 61831013. The associate editor coordinating the review of this article and approving it for publication was C. Huang. *(Corresponding author: Feifei Gao.)*

Zhe Ma and Feifei Gao are with the State Key Laboratory of Intelligent Technologies and Systems, Beijing National Research Center for Information Science and Technology, Department of Automation, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China (e-mail: maz16@mails.tsinghua.edu.cn; feifeigao@ieee.org).

Wen Wu and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: w77wu@uwaterloo.ca; sshen@uwaterloo.ca).

Mengnan Jian is with the State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China, and also with the Algorithm Department, Wireless Product R&D Institute, ZTE Corporation, Shenzhen 518057, China (e-mail: jian.mengnan@zte.com.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3108666.

Digital Object Identifier 10.1109/TWC.2021.3108666

end-to-end communication. The core 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 a holistic manner. The idea was first pioneered in [30], and has subsequently led to many extensions [31]–[33]. However, existing works on end-to-end communication are generally based on the traditional autoencoder, where the encoder converts its input to a discrete latent space that may lack interpretable and exploitable structures [34]–[35]. The variational autoencoder can enhance the traditional autoencoder in terms of interpretability by imposing a prior distribution on latent variables and forming a smooth latent space, referred to as "disentanglement learning" in the literature [36]–[37]. Besides, disentanglement learning can significantly improve the downstream tasks (e.g., input reconstruction) and is useful for a large variety of domains [38]. Considering these appealing properties, we adopt a variational autoencoder network for the joint optimization design of MCD and MUD.

There are several works considering the application of variational autoencoder in communication systems. In [39], a deep learning method is designed for device activity detection in mMTC under imperfect CSI using variational-autoencoder. In [40], an unsupervised neural network-based algorithm is proposed for blind channel equalization using the method of variational autoencoder. However, the original variational autoencoder adopted in these existing works cannot learn the complicated coupling relation among signal streams, and thus may not be suitable for the considered GF-NOMA system. Additionally, considering that the user heterogeneity (e.g., different transmission rates) may lead to unbalanced training performance among users [41], a multi-task prioritizing strategy is needed to guarantee training fairness. To address the issues above and abandon the "black-box" of machine learning [42], we carry out theoretical analysis based on information theory, revealing the rationale behind the network structure. To the best of our knowledge, this is the first attempt to investigate the joint optimization of MCD and MUD using DL with theoretical analysis. Simulation results validate that the proposed method outperforms the conventional methods in terms of both detection accuracy and computational complexity. The main contributions can be summarized as follows.

- We formulate the joint optimization problem using variational inference, based on which the loss function is derived to ensure that the proposed network can be trained to approach the optimality. On this basis, we provide a theoretical analysis that reveals the learning mechanism and offers valuable insights for the detailed network structure design.
- We analyze the loss function from an information-theoretic perspective, unveiling that the detection processes of different users are mutually conflicted and correlated.
- We develop a DL-based method, namely multi-task variational autoencoder (Mul-VAE), to jointly optimize the MCD and MUD for the GF-NOMA system. A multi-task learning structure, i.e., the sluice networ [43], is incorporated in the variational autoencoder to handle the mutually conflicting but related detection processes among different users.
- We leverage the idea of focal loss [44] to devise a multi-task prioritizing strategy, where the training fairness is guaranteed by adaptively adjusting the weight coefficient of each task.

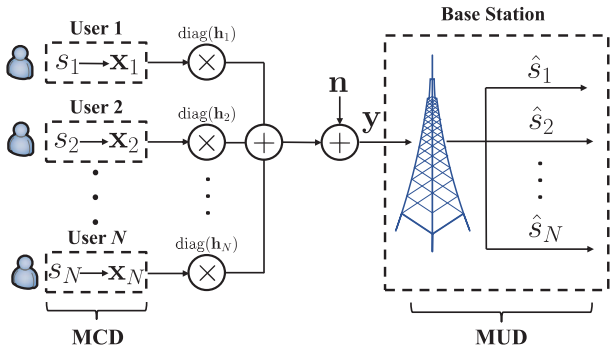


Fig. 1. System model of the mMTC uplink scenario.

The remainder of the paper is organized as follows. In Section II, we formulate the joint optimization problem of MCD and MUD. In Section III, we elaborate the proposed Mul-VAE network. Simulation results are presented in Section IV, and conclusions are made in Section V.

Notations: We use normal lower-case, bold lower-case, and bold upper-case letters to denote scalars, vectors, and matrices, respectively. For matrix $\mathbf{X}$, $\mathbf{X}^T$ denotes its transpose, tr($\mathbf{X}$) denotes its trace, and det($\mathbf{X}$) denotes its determinant. For vector $\mathbf{x}$, $||\mathbf{x}||$ denotes its l_2-norm, and diag($\mathbf{x}$) denotes the diagonal matrix with the diagonal specified by $\mathbf{x}$. $[x_n]_{n=1}^N$ denotes the N-dimensional vector where x_n is the nth element. $\mathbb{R}^{M\times N}$ and $\mathbb{C}^{M\times N}$ denote the $M \times N$ dimensional real space and complex space, respectively. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the multivariate Gaussian and complex Gaussian distributions with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, respectively.

II. System Model and Problem Formulation

A. System Model

As illustrated in Fig. 1, we consider an uplink mMTC network consisting of N users and one BS, where all terminals are equipped with a single antenna. A GF-NOMA scheme is adopted, where the users can access the BS without a prior scheduling assignment. We assume that the nth user, $\forall n \in \{1, 2, \cdots, N\}$, wishes to send a message $s_n \in \{1, 2, \cdots, 2^{R_n}\}$ to the BS at a rate R_n. Note that R_n is not necessarily equal to each other, owing to the heterogeneity of mMTC users [46]. The message s_n is transmitted over K orthogonal resources (e.g., subcarriers or OFDM tones), and thus is mapped 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 $||\mathbf{x}_n|| \leq K$, where K is usually smaller than N in the GF-NOMA scheme. The received signal at the BS is

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

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 additive white Gaussian noise (AWGN) distributed as $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_K)$ with $\mathbf{I}_K$ being a $K \times K$ identity matrix. It is assumed that the transmission interval is shorter than the channel coherence time in each resource.

Upon the reception of $\mathbf{y}$, the BS tries to recover all user messages through the MUD process. The communication procedure of the nth user can be expressed as the cascade of the MCD function and MUD function, i.e.,

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

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

B. Problem Formulation

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 $\mathbb{P}(\hat{\mathbf{s}} \neq \mathbf{s})$, where $\hat{\mathbf{s}} = [\hat{s}_1, \cdots, \hat{s}_N]^T$ and $\mathbf{s} = [s_1, \cdots, s_N]^T$. We start the formulation from the maximum likelihood estimation, which yields the optimal MUD by maximizing the following marginal likelihood

$$\begin{aligned}[g_n^*]_{n=1}^N &= \underset{g_n, \forall n\in\mathfrak{N}}{\arg\max}\ \log P(\mathbf{s}; [g_n]_{n=1}^N) \\ &= \underset{g_n, \forall n\in\mathfrak{N}}{\arg\max}\ \log \int P(\mathbf{s}|\mathbf{y}; [g_n]_{n=1}^N) P(\mathbf{y})\, d\mathbf{y}, \end{aligned} \tag{3}$$

where $\mathfrak{N} = \{1, 2, \cdots, N\}$, $P(\mathbf{s}; [g_n]_{n=1}^N)$ is the probability distribution of the recovered message parameterized by $[g_n]_{n=1}^N$, $P(\mathbf{y})$ is the probability distribution of $\mathbf{y}$, and $P(\mathbf{s}|\mathbf{y}; [g_n]_{n=1}^N)$ is the posterior probability that $[g_n]_{n=1}^N$ correctly recovers $\mathbf{s}$ given $\mathbf{y}$.

By the variational inference, we have the following theorem that transforms (3) into the maximum of the *variational lower bound* over $[f_n]_{n=1}^N$ and $[g_n]_{n=1}^N$.

Theorem 1: The marginal likelihood $\log P(\mathbf{s}; [g_n]_{n=1}^N)$ can be rewritten as

$$\begin{aligned}&\log P(\mathbf{s}; [g_n]_{n=1}^N) \\ &\geq \log P(\mathbf{s}; [g_n]_{n=1}^N) - KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y}|\mathbf{s}; [g_n]_{n=1}^N)) \\ &= \mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N) \\ &= \sum_{n=1}^{N} \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[\log P(s_n|\mathbf{y}; g_n)] \\ &\quad - KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y})), \end{aligned} \tag{4}$$

where $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ is the *variational lower bound* [47], $Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)$ is the parameterized (by $[f_n]_{n=1}^N$) conditional probability distribution of $\mathbf{y}$ given $\mathbf{s}$, and $KL(p(z)||q(z)) = \int p(z) \log(p(z)/q(z))\, dz$ is the Kullback-Leibler divergence (KLD) that measures the similarity between two distributions.

Proof: Please refer to Appendix A.

Remark 1: From Theorem 1, we see that $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ is a lower bound of $\log P(\mathbf{s}; [g_n]_{n=1}^N)$, and $KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y}|\mathbf{s}; [g_n]_{n=1}^N))$ determines the gap between $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ and $\log P(\mathbf{s}; [g_n]_{n=1}^N)$. Meanwhile, it can also be observed that maximizing $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ will concurrently maximize the marginal likelihood $\log P(\mathbf{s}; [g_n]_{n=1}^N)$ and minimize the gap term $KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y}|\mathbf{s}; [g_n]_{n=1}^N))$, which implies the equivalence between the maximization of $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ and $\log P(\mathbf{s}; [g_n]_{n=1}^N)$. It is further noted that $Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)$ can be regarded as the MCD process that maps s_n to $\mathbf{x}_n, \forall n \in \mathfrak{N}$, and then generates $\mathbf{y}$, as there exists a deterministic mapping between $\mathbf{y}$ and $\mathbf{x}$ in (1). Similarly, $P(s_n|\mathbf{y}; g_n)$ can be considered as the nth MUD process, which extracts s_n from $\mathbf{y}$. Therefore, we readily see that (3) has been converted into the maximum of $\mathcal{L}([f_n]_{n=1}^N, [g_n]_{n=1}^N)$ over all possible MCD and MUD mappings. Taking the user weight into account, we can formulate the joint optimization problem of MCD and MUD as

$$\begin{aligned}\mathcal{P}1: \ \underset{f_n, g_n, \forall n\in\mathfrak{N}}{\arg\min}\ &-\sum_{n=1}^{N} w_n \mathbb{E}_{Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N)}[\log P(s_n|\mathbf{y}; g_n)] \\ &+ KL(Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) || P(\mathbf{y})), \\ \text{s.t. } \mathbf{y} &= \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n)\mathbf{x}_n + \mathbf{n}, \end{aligned} \tag{5}$$

where w_n is the weight coefficient for user n. These weight coefficients can balance the training performance among users by adaptively adjusting their values. The detailed design of w_n is presented in Section III-D and thus is omitted here for brevity.

However, it is non-trivial to solve (5) analytically, due to the intricately coupled f_n and g_n as well as their infinite searching spaces. To solve the problem, we take a DL-based method to approximate the optimal solution, by tapping the strong nonlinear approximation capability of DNNs, which will be elaborated in the next section.

III. DL-Based Joint Optimization for GF-NOMA

This section proposes a variational autoencoder-based network, namely Mul-VAE, to jointly optimize MCD and MUD for GF-NOMA. The key idea is to parameterize (5) as a variational autoencoder, where the encoder and decoder are trained to mimic the optimal MCD and MUD, respectively. We first present the general architecture of the proposed Mul-VAE and derive the corresponding loss function. Then, we analyze the loss function from an information-theoretic perspective, which offers insights into the design of detailed network structure. On this basis, we scrutinize the carefully designed structure of each individual module. After that, we propose a multi-task prioritizing strategy to address the training unfairness issues among users and present the training algorithm.

A. General Architecture

As depicted in Fig. 2, the proposed Mul-VAE network consists of a probabilistic encoder denoted by $F_{\mathbf{\Theta}}(\cdot)$, and a probabilistic 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 $[g_n]_{n=1}^N$,

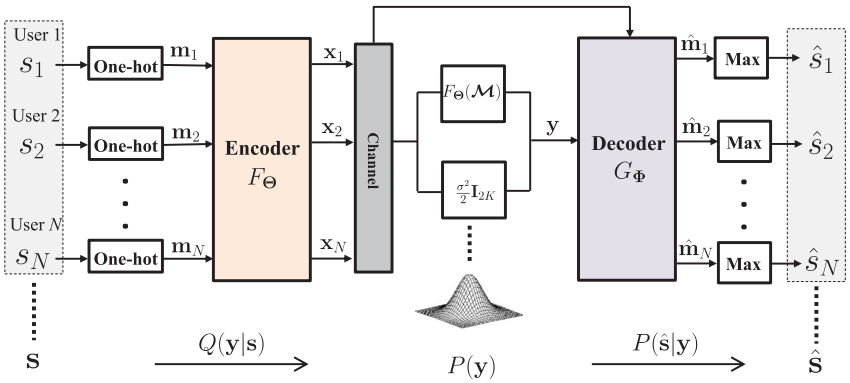


Fig. 2. General architecture of the proposed Mul-VAE.

respectively. In essence, the encoder approximates the optimal MCD by learning the optimal mapping between user messages $\mathbf{s}$ and multidimensional symbols $\mathbf{x}_n, \forall n \in \mathfrak{N}$, while the decoder mimics the optimal MUD by learning the optimal mapping from $\mathbf{y}$ to $\mathbf{s}$.

At the encoder, we apply the one-hot encoding to represent user messages, as it can simplify (5) into a classification problem. In one-hot encoding, each message $s_n \in \{1, 2, \cdots, 2^{R_n}\}$ is represented by a 2^{R_n}-dimensional vector $\mathbf{m}_n \in \{0, 1\}^{2^{R_n}}$, where the s_nth element is 1 and the others are all 0.

The one-hot encoded messages are then forwarded to the encoder to generate the multidimensional constellations $\mathbf{x}_n = F_{\mathbf{\Theta}}(\mathbf{m}_n), n \in \mathfrak{N}$. To facilitate the learning process of Mul-VAE, we convert $\mathbf{x}_n$ to its real signal version by concatenating its real and imaginary parts. Therefore, $F_{\mathbf{\Theta}}(\mathbf{m}_n)$ is a real-valued vector with $2K$ dimension, and the corresponding constellation is determined as[1]

$$\mathbf{x}_n = F_{\mathbf{\Theta}}(\mathbf{m}_n)_{(1:K)} + \sqrt{-1}F_{\mathbf{\Theta}}(\mathbf{m}_n)_{(K+1:2K)}, \tag{6}$$

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

$$\mathbf{y} = \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n) F_{\mathbf{\Theta}}(\mathbf{m}_n) + \mathbf{n}. \tag{7}$$

Consequently, we can approximate the MCD process in (5) by

$$Q(\mathbf{y}|\mathbf{s}; [f_n]_{n=1}^N) = \mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K}), \tag{8}$$

where $\boldsymbol{\mathcal{M}} = [\mathbf{m}_1, \cdots, \mathbf{m}_N]$, $F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) = \sum_{n=1}^{N} \text{diag}(\mathbf{h}_n) F_{\mathbf{\Theta}}(\mathbf{m}_n)$, and $\mathbf{I}_{2K}$ is a $2K \times 2K$ identity matrix. Note that we take the channel as the output layer of the proposed encoder, such that the mean value and variance of $\mathbf{y}$ involved in (8) can be directly accessed [34].

At the decoder, we assume that the channel state information (CSI) is perfectly known since it can be obtained by the BS through pilot-based channel estimation methods [48]. Hence, both the CSI and received signal $\mathbf{y}$ can be fed to the decoder as the input. The last layer of the decoder 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 $||\hat{\mathbf{m}}_n||_1 = 1$, $n \in \mathfrak{N}$. The softmax layer converts the output of the decoder into the same dimension with the one-hot encoded user messages, such that the loss function can be directly computed based on the input and output of Mul-VAE. Accordingly, the MUD process can be approximated as a categorical distribution

$$P(s_n|\mathbf{y}; g_n) = P(\mathbf{m}_n|\mathbf{y}; \mathbf{\Phi}) = \prod_{i=1}^{2^{R_n}} \hat{\mathbf{m}}_{n_i}^{\mathbf{m}_{n_i}} = \prod_{i=1}^{2^{R_n}} G_{\mathbf{\Phi}}(\mathbf{y}, \mathbf{H})_{n_i}^{\mathbf{m}_{n_i}}, \tag{9}$$

where $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_N\}$ and $G_{\mathbf{\Phi}}(\cdot)_{n_i}$ is the n_ith element of the decoder 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 (8) and (9) into (5), the loss function can be determined as

$$\begin{aligned}
&\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) \\
&= -\sum_{n=1}^{N} w_n \mathbb{E}_{\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})} \left[\sum_{i=1}^{2^{R_n}} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(\mathbf{y}, \mathbf{H})_{n_i}) \right] \\
&\quad + KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})||P(\mathbf{y})) \\
&= \mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi}) + \mathcal{L}_{KL}(\mathbf{\Theta}),
\end{aligned} \tag{10}$$

where

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

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

is the expected reconstruction loss and $\mathcal{L}_{KL}(\mathbf{\Theta}) = KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})|P(\mathbf{y}))$ is the regularization term that makes $\mathbf{y}$ follow a prior distribution. In the following, we respectively derive these two terms in detail, based on which we obtain the loss function.

1) Derivation of $\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi})$: It is noted that computing $\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi})$ involves a sampling operation, which stunts the backpropagation since sampling operation is non-differentiable. To circumvent this problem, we adopt the reparameterization method in [34]. Its core idea is to randomly draw a sample ϵ from a unit Gaussian distribution, 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 [34]

$$\begin{aligned}\mathcal{L}_R(\mathbf{\Theta}, \mathbf{\Phi}) &= -\sum_{n=1}^{N} w_n \sum_{s=1}^{S}\sum_{i=1}^{2^{R_n}} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) + \boldsymbol{\epsilon}^{(s)}, \mathbf{H})_{n_i}) \\ &= \sum_{n=1}^{N} w_n \mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}), \end{aligned} \tag{11}$$

where $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) = -\sum_{s=1}^{S}\sum_{i=1}^{2^{R_n}} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) + \boldsymbol{\epsilon}^{(s)}, \mathbf{H})_{n_i})$, S is the number of samples, and $\boldsymbol{\epsilon}^{(s)} \sim \mathcal{N}(0, \frac{\sigma^2}{2}\mathbf{I}_{2K})$. It is found in [34] that when batch size is large enough (e.g., larger than 100), the sample number S can be set to 1. Therefore, $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi})$ can be simplified as $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) = -\sum_{i=1}^{2^{R_n}} \mathbf{m}_{n_i} \log(G_{\mathbf{\Phi}}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) + \boldsymbol{\epsilon}^{(s)}, \mathbf{H})_{n_i})$, which is exactly the cross-entropy loss that is widely used in DL-based applications.

2) Derivation of $\mathcal{L}_{KL}(\mathbf{\Theta})$: Another critical issue is to determine $P(\mathbf{y})$, 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 $\mathbf{y}$.[2] Therefore, we set $P(\mathbf{y}) = \mathcal{N}(\mathbf{p}_R, \frac{\sigma^2}{2}\mathbf{I}_{2K})$ and derive $\mathcal{L}_{KL}(\mathbf{\Theta})$ as

$$\begin{aligned}\mathcal{L}_{KL}(\mathbf{\Theta}) &= KL(\mathcal{N}(\mathbf{y}|F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}), \frac{\sigma^2}{2}\mathbf{I}_{2K})||\mathcal{N}(\mathbf{p}_R, \frac{\sigma^2}{2}\mathbf{I}_{2K})) \\ &= \frac{2}{\sigma^2}(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - \mathbf{p}_R)^T(F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}}) - \mathbf{p}_R), \end{aligned} \tag{12}$$

where $\mathbf{p}_R = [\sqrt{P_R}, \cdots, \sqrt{P_R}]^T \in \mathbb{R}^{2K\times 1}$ and P_R is the average received signal power. It should be mentioned that $\mathcal{L}_{KL}(\mathbf{\Theta})$ can be approximated by 0 when $N \to \infty$. This is because if the distribution of user messages, i.e., $P(\boldsymbol{\mathcal{M}})$, is fixed, then $F_{\mathbf{\Theta}}(\boldsymbol{\mathcal{M}})$ will converge to $\mathbf{p}_R$ when $N \to \infty$ according to the law of large numbers.

3) Loss Function: Based on (11), (12), and the above discussions, the loss function can be rewritten as

$$\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi}) = \sum_{n=1}^{N} w_n \mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}). \tag{13}$$

Remark 2: 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), (13) is different to the cross-entropy loss, which implies that the cross-entropy loss function is no longer optimal. That is, classic loss functions may not be suitable in some wireless communication scenarios, and thus more insights should be offered into the loss function design.

4) Information-Theoretic Analysis: Having presented the overall architecture of the proposed Mul-VAE, we analyze the loss function based on information theory, which provides valuable insights for the network structure design.

According to the definition of cross-entropy [50], we can rewrite $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi})$ as

$$\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) = -\sum_{i=1}^{2^{R_n}} \mathbf{m}_{n_i} \log(\hat{\mathbf{m}}_{n_i}) = -P(\mathbf{m}_n) \log P(\hat{\mathbf{m}}_n), \tag{14}$$

where $P(\mathbf{m}_n)$ and $P(\hat{\mathbf{m}}_n)$ are the distributions of $\mathbf{m}_n$ and $\hat{\mathbf{m}}_n$, respectively. Since $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi})$ is averaged over all possible user messages and channel outputs during the training process, we can further express (14) as

$$\begin{aligned}\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}) &= \mathbb{E}_{P(\mathbf{m}_n, \mathbf{y}; \mathbf{\Theta})}[-\log P(\hat{\mathbf{m}}_n|\mathbf{y}; \mathbf{\Phi})] \\ &\overset{(a)}{=} H(\mathbf{m}_n) - I(\mathbf{m}_n, \mathbf{y}; \mathbf{\Theta}) \\ &\quad + \mathbb{E}_{P(\mathbf{y})}[KL(P(\mathbf{m}_n|\mathbf{y}; \mathbf{\Theta})||P(\hat{\mathbf{m}}_n|\mathbf{y}; \mathbf{\Phi}))] \\ &\overset{(b)}{=} H(\mathbf{m}_n) - I(\mathbf{m}_n, \mathbf{y}; \mathbf{\Theta}) \\ &\quad + \mathbb{E}_{P(\mathbf{y})}[KL(\int P(\mathbf{m}_n|\mathbf{x}; \mathbf{\Theta}) P(\mathbf{x}|\mathbf{y})\, d\mathbf{x}|| \\ &\quad \int P(\hat{\mathbf{m}}_n|\mathbf{x}; \mathbf{\Phi}) P(\mathbf{x}|\mathbf{y})\, d\mathbf{x})], \end{aligned} \tag{15}$$

where $H(\mathbf{m}_n)$ is the entropy of $\mathbf{m}_n$, and $I(\mathbf{m}_n, \mathbf{y}; \mathbf{\Theta})$ is the mutual information (MI) between $\mathbf{m}_n$ and $\mathbf{y}$. The detailed derivation of (a) can be found in [51], and (b) holds due to the probability factorization. We mainly focus on the last two right-hand side (RHS) terms of (15), as $H(\mathbf{m}_n)$ is a constant due to the fixed distribution of $\mathbf{m}_n$. Based on the analysis in (15), we specify the following three observations, which serves as the underlying motivation for the detailed structure design in Section III-C.

Observation 1 - Conflicting Term: The second RHS term $I(\mathbf{m}_n, \mathbf{y}; \mathbf{\Theta})$ causes conflicts among different users. Considering an example with two users, both $\mathcal{L}_{R_1}(\mathbf{\Theta}, \mathbf{\Phi})$ and $\mathcal{L}_{R_2}(\mathbf{\Theta}, \mathbf{\Phi})$ should be minimized to reach the minimum $\mathcal{L}(\mathbf{\Theta}, \mathbf{\Phi})$. According to (15), minimizing $\mathcal{L}_{R_1}(\mathbf{\Theta}, \mathbf{\Phi})$ requires maximizing $I(\mathbf{m}_1, \mathbf{y}; \mathbf{\Theta})$, while minimizing $\mathcal{L}_{R_2}(\mathbf{\Theta}, \mathbf{\Phi})$ requires maximizing $I(\mathbf{m}_2, \mathbf{y}; \mathbf{\Theta})$. These two objectives are contradictory since maximizing $I(\mathbf{m}_1, \mathbf{y}; \mathbf{\Theta})$ does not necessarily lead to maximizing $I(\mathbf{m}_2, \mathbf{y}; \mathbf{\Theta})$, which indicates that all $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}), n \in \mathfrak{N}$ cannot be optimized simultaneously.

Observation 2 - Common Distribution Term: It can be observed from the third RHS term that all $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi}), n \in \mathfrak{N}$ share a common distribution term $P(\mathbf{x}|\mathbf{y})$, which implies that a preprocessing module should be deployed at the decoder to obtain this common distribution.

Observation 3 - Related Individual Distribution Term: Besides the common distribution term, each $\mathcal{L}_{R_n}(\mathbf{\Theta}, \mathbf{\Phi})$ also

[2] A similar interpretation was provided in [49], where Shapiro-Wilk test was employed to measure the normality of constellation points.

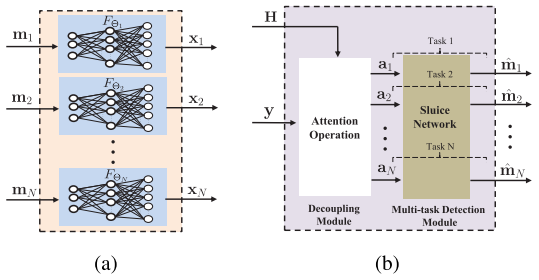


Fig. 3. Detailed network structures of (a) the encoder and (b) the decoder.

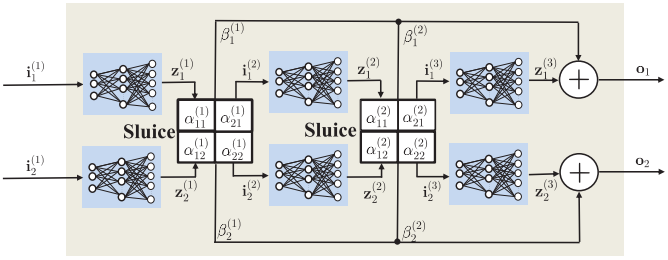


Fig. 4. Architecture of a three-layer sluice network.

has a specific distribution term $P(\mathbf{m}_n|\mathbf{x};\mathbf{\Theta})$, indicating that different MUD processes are associated. Accordingly, the proposed network should be able to exploit the relationship among different MUD processes.

C. Detailed Network Structure

Given the above observations, we now summarize the design principles: (i) According to **Observation 1** and **Observation 3**, we adopt a novel multi-task learning structure to handle the mutually conflicting yet related learning process. It should be mentioned that we only incorporate multi-task learning at the decoder, as the encoder should be designed in a distributed manner due to the independence among users; (ii) Considering **Observation 2**, we further develop a decoupling module at the decoder to preprocess the received signal. It is worth noting that $P(\mathbf{x}|\mathbf{y})$ is essentially a decompressing process since K is smaller than N in the GF-NOMA system. Therefore, the goal of the decoupling module is to decompose the condensed signal $\mathbf{y}$ into a high-dimensional feature vector $\mathbf{x}$. Based on these two design principles, the detailed network structure is elaborated as follows.

1) Structure of the Encoder: 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 $F_{\mathbf{\Theta}_n} : \mathbf{m}_n \rightarrow \mathbf{x}_n, n \in \mathfrak{N}$, 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

$$\mathbf{z}_n^l = \mathfrak{f}_n^l(\boldsymbol{W}_n^l \mathbf{z}_n^{l-1} + \boldsymbol{b}_n^l), \tag{16}$$

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., $\mathfrak{f}_n^l(x) = (1-e^{-2x})/(1+e^{-2x})$) is adopted as the activation function in all the hidden layers. As a result, the overall input-output affine of $F_{\mathbf{\Theta}_n}$ is

$$F_{\mathbf{\Theta}_n}(\mathbf{m}_n) = \mathfrak{f}_n^{L-1}(\mathfrak{f}_n^{L-2}(\cdots(\mathfrak{f}_n^1(\mathbf{m}_n;\mathbf{\Theta}_n^1),\cdots)\mathbf{\Theta}_n^{L-2});\mathbf{\Theta}_n^{L-1}). \tag{17}$$

2) Structure of the Decoder: As illustrated in Fig. 3, the decoder comprises two parts: the decoupling module and the multi-task detection module.

a) Decoupling module: The decoupling module employs the attention mechanism [52], i.e., mapping a query and a set of key-value pairs to the output, for the signal decomposition. By taking $\mathbf{y}$ as the query and $\text{diag}(\mathbf{h}_n), n \in \mathfrak{N}$ as the key-value pairs, we can decompress $\mathbf{y}$ to a $2KN$-dimensional vector as

$$\mathbf{a}_n = \mathfrak{f}_A(\text{diag}(\mathbf{h}_n)^\mu \mathbf{y}), \tag{18}$$

where $\mathbf{a}_n$ is the nth attention vector, $\mathfrak{f}_A$ is the activation function for the attention operation, and μ is the product parameter. The attention vectors $\mathbf{a}_n, n \in \mathfrak{N}$ are then forwarded to the multi-task detection module.

b) Multi-task detection module: The multi-task detection module employs a multi-task learning structure, where each MUD process is taken as one single *task*. During the jointly training, each task shares its learned information with other tasks [45]. In this way, the multi-task detection module can exploit the relationship among different MUD processes as an inductive bias, which eases the simultaneous optimization of all the conflicting but related MUD processes. Particularly, we adopt the sluice network that learns the optimal latent connections and pathways among tasks, to avoid the costly searching for potentially optimal relational parameters [43].

To make the sluice network more clear, a three-layer sluice network for the case of two users is sketched in Fig. 4, which consists of two task-specific input layers, two task-specific output layers, two sluices, and three DNN layers per task. The sluice network operates in the following procedure: (i) The input vectors are fed into the first DNN layers for nonlinear transformation, denoted as $\mathbf{z}_n^{(1)} = F_{DNN}^{(1)}(\mathbf{i}_n^{(1)}), n = 1, 2$, where F_{DNN} is the overall input-output affine of DNN. The details of F_{DNN} have been presented in Section III-C and thus are omitted here for brevity; (ii) The outputs of the first DNN layers are then passed to the second DNN layers. The traffic of information is mediated by a set of parameters $\boldsymbol{\alpha}^{(1)} = \{\alpha_{11}^{(1)}, \alpha_{12}^{(1)}, \alpha_{21}^{(1)}, \alpha_{22}^{(1)}\}$ and $\boldsymbol{\beta}^{(1)} = \{\beta_1^{(1)}, \beta_2^{(1)}\}$, which is similar to the way a sluice controls the flow of water. Specifically, the second DNN layers receive the combination of the outputs of two first DNN layers weighted by $\boldsymbol{\alpha}$, i.e., $\mathbf{i}_1^{(2)} = \alpha_{11}^{(1)}\mathbf{z}_1^{(1)} + \alpha_{21}^{(1)}\mathbf{z}_2^{(1)}$ and $\mathbf{i}_2^{(2)} = \alpha_{22}^{(1)}\mathbf{z}_2^{(1)} + \alpha_{12}^{(1)}\mathbf{z}_1^{(1)}$. Interestingly, if $\alpha_{11}^{(1)} = \alpha_{22}^{(1)} = 1$ and $\alpha_{12}^{(1)} = \alpha_{21}^{(1)} = -1$, the sluicing process is similar to the SIC detection; (iii) The previous process is then repeated in the second sluice and third DNN layer; (iv) Finally, the outputs at various depths are mediated through $\boldsymbol{\beta}$ and superposed as the outputs of the sluice network, i.e., $\mathbf{o}_1 = \beta_1^{(1)}\mathbf{z}_1^{(1)} + \beta_1^{(2)}\mathbf{z}_1^{(2)} + \mathbf{z}_1^{(3)}$ and

$\mathbf{o}_2 = \beta_2^{(1)}\mathbf{z}_2^{(1)} + \beta_2^{(2)}\mathbf{z}_2^{(2)} + \mathbf{z}_2^{(3)}$. It is worth noting that both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are trainable, which allows the network to learn the appropriate architecture and amount for sharing among tasks.

D. Multi-Task Prioritizing

The user heterogeneity usually results in unbalanced task difficulty. For example, it is more challenging to detect the symbols for high rate users than low rate users. The imbalance in task difficulty can lead to unnecessary emphasis on easier tasks, thus neglecting and stunting progress on difficult tasks [41]. Meanwhile, even when tasks have the same difficulty (e.g., the same rate for users), some tasks are prioritized while other tasks are ignored occasionally, owing to the random fluctuation of the gradient-based optimizer. The unbalanced computational resource allocation causes unfairness among various tasks or even leads to divergence.

To address the above issues, we design a task prioritizing strategy, whose *core idea* is to speed up or slow down the backpropagated gradients of tasks by adjusting their weights according to the learning performance. To this end, we apply the *focal loss* as task weights w_n, which is defined as [44]

$$FL_n(\kappa_n) = -(1-\kappa_n)^{\gamma}\log(\kappa_n). \tag{19}$$

Here $\kappa_n \in (0,1]$ is the key performance indicator (KPI) of task n, i.e., the classification accuracy of the nth MUD process, and γ is a hyper-parameter adjusting the rate at which tasks are down-weighted.

Remark 3: It can be noted that when a task is well learned, i.e., $\kappa_n \to 1$, both $(1-\kappa_n)^{\gamma}$ and $\log(\kappa_n)$ go to 0, and thus the focal loss is diminished. When a task is poorly learned, i.e., $\kappa_n \to 0$, the factor $(1-\kappa_n)^{\gamma}$ is near 1 and $\log(\kappa_n) \to \infty$, and thus the focal loss is magnified. Therefore, the focal loss can adaptively adjust itself to be inversely proportional to the task performance, reducing the loss contribution from well-learned tasks while increasing the importance of correcting poorly-learned tasks.

However, the focal loss cannot be directly applied to Mul-VAE, since only the training loss can be obtained during the training process. To tackle this issue, we derive an approximate classification accuracy from the cross entropy loss, which yields the following theorem.

Theorem 2: When the batch size goes to infinity, the classification accuracy κ can be approximated by

$$\kappa \approx \frac{e^{-\mathcal{L}_C}}{0.7357}, \tag{20}$$

where $\mathcal{L}_C$ is the cross entropy loss.

Proof: Please refer to Appendix B.

Remark 4: From Theorem 2, one may want to train the network utilizing a large batch size, as it should lead to a better approximation of the true classification accuracy. However, using extremely large batch sizes usually degrades the training stability and generalization performance of neural networks, as experimentally observed in [53]. Therefore, the batch size controls a tradeoff between the precision of the approximation and the learnability of the network. We empirically determine the batch size, which will be discussed in Section IV-A. The theoretical discussion of the batch size is beyond the scope of this paper and will be left in future work.

Algorithm 1 Training Algorithm

Input: Training dataset D, batch size B, symbol dimension K, orthogonal resource number N, prioritizing hyper parameter γ, and attention parameter μ.
Output: Trained parameter $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$.
$\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$ $\leftarrow$ Random initialization.
Repeat

- D_B $\leftarrow$ Randomly draw a batch with B training samples from D;
- $\mathcal{L}_B \leftarrow \mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi}; D_B)$ (Calculate the average loss function based on D_B according to (21));
- $\nabla_{\boldsymbol{\Theta}}, \nabla_{\boldsymbol{\Phi}} \leftarrow \nabla_{\boldsymbol{\Theta},\boldsymbol{\Phi}}\mathcal{L}_B$ (Calculate gradients based on $\mathcal{L}_B$);
- $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$ $\leftarrow$ Update parameters using gradients $\nabla_{\boldsymbol{\Theta}}, \nabla_{\boldsymbol{\Phi}}$ through the Adam optimizer;

Until convergence of parameters ($\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$).
Return $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$

Integrating (13), (19), and (20), the loss function with multi-task prioritization is given by

$$\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi}) = -\log(0.7357)\sum_{n=1}^{N}\left(1-\frac{e^{-\mathcal{L}_{R_n}(\boldsymbol{\Theta},\boldsymbol{\Phi})}}{0.7357}\right)^{\gamma}\mathcal{L}_{R_n}^2(\boldsymbol{\Theta},\boldsymbol{\Phi}). \tag{21}$$

E. Training Algorithm

This section presents the training algorithm, through which the proposed Mul-VAE can be trained to approach the optimal MCD and MUD. When training the Mul-VAE network, we aim to find the optimal network parameter sets $\boldsymbol{\Theta}^*$ and $\boldsymbol{\Phi}^*$ that minimize the loss function $\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi})$ using the training dataset. The training dataset is synthetically generated by randomly sampling a sequence of data points from the one-hot vector. Denoting $D_n = [\mathbf{m}_n^{(t)}]_{t=1}^{T}$ as the training dataset for user n, the whole training dataset D is given by

$$D = \{D_1, \cdots, D_N\} = [\boldsymbol{\mathcal{M}}^{(t)}]_{t=1}^{T} \in \mathbb{R}^{\sum_{n=1}^{N} 2^{R_n} \times T}, \tag{22}$$

where T is the size of the training dataset. Note that both users and BS can access the training dataset by utilizing pseudorandom number generators initialized with the same seed [33]. Hence, we can apply D as both the input data and the output label of Mul-VAE, thereby eliminating the need for human labeling effort.

Mul-VAE is trained epoch by epoch in an end-to-end manner. While within an epoch, a batch $D_B = [\boldsymbol{\mathcal{M}}^{(t)}]_{t=1}^{B}$ of size B is randomly drawn out of D and fed into the encoder to generate multidimensional symbols. The encoded symbols are then sent over the channel, after which the decoder receives the composite signal $\mathbf{y}$ and outputs the probability vectors $[\hat{\boldsymbol{\mathcal{M}}}^{(t)}]_{t=1}^{B}$ through the softmax layer. The loss function is computed and averaged based on the output label and the output vector as $\mathcal{L}_B(\boldsymbol{\Theta},\boldsymbol{\Phi}) = \frac{1}{B}\sum_{t=1}^{B}\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi};\boldsymbol{\mathcal{M}}^{(t)},\hat{\boldsymbol{\mathcal{M}}}^{(t)})$. Next, the gradients $\nabla_{\boldsymbol{\Theta}}$ and $\nabla_{\boldsymbol{\Phi}}$ are calculated based on $\mathcal{L}_B$ using the backpropagation algorithm, and the parameter sets

are updated simultaneously with the Adam optimizer. This training process is carried out until the convergence of $\mathbf{\Theta}$ and $\mathbf{\Phi}$ (e.g., the training loss stops decreasing). The whole process is summarized in **Algorithm 1.**

IV. Performance Evaluation

This section presents the simulation results and corresponding computational complexity analysis to demonstrate the superiority of Mul-VAE over competing algorithms.

A. Simulation Setup

Simulation setup is based on the typical NOMA scenario [9], where $N = 6$ users share $K = 4$ orthogonal resources. Considering users have heterogeneous transmission rates, we assume that 3 of the 6 users (User1, User2, and User3) send their messages at $R = 1$ bit/s, while the other 3 users (User4, User5, and User6) transmit at $R = 2$ bits/s.[3] The encoder of Mul-VAE consists of 6 DNNs. For each DNN, there are 4 hidden layers with $2^{(R+3)}$ neurons. The decoder of Mul-VAE has two parts, namely the decoupling module and the multi-task detection module. For the decoupling module, we set the product hyper-parameter as $\mu = -1$ and the activation function as an identity function. For the multi-task detection module, we adopt a two-layer sluice network, where each DNN has 5 hidden layers with $2^{(R+3)}$ neurons. Meanwhile, the multi-task prioritizing strategy is employed with the prioritizing hyper-parameter set as $\gamma = 2$.

We train Mul-VAE with $80,000$ training samples for $5,000$ epochs and test with $20,000$ data samples. The learning rate is set to be 2×10^{-5} and the batch size is 500. The network is Xavier initialized and trained at the signal to noise ratio (SNR) of 2 dB but is tested at a wide range of SNR. All the parameters are empirically determined using the general workflow, where the network starts with relative small parameters and increases the values until the learning performance cannot be further promoted. Mul-VAE is implemented on an x86 PC with one Nvidia GeForce GTX 1080 Ti graphics card, and Pytorch 1.1.0 is employed as the backend. For comparison, the performance of the following three benchmark algorithms is also evaluated.

- **SCMA with MPA detector (SCMA-MPA):** In SCMA-MPA, we apply the user-to-resource mapping and codebook generating method in [54] and set the iteration number N_{iter} of MPA to be 5.
- **Multiuser shared access with MMSE-SIC detector (MUSA-SIC):** In MUSA-SIC, the QPSK modulation is adopted for high rate users, and the BPSK modulation is used for low rate users. The complex spreading code is generated based on a 3-ary set $\{1, 0, -1\}$.
- **Conventional end-to-end network (E2E):** In conventional E2E network [30]– [31], the DNNs in both the encoder and the decoder have the same width and depth as those in Mul-VAE.

[3]It should be noted that the proposed network can be extended to higher rates and larger networks with slight modifications of the network structure.

B. Evaluation of the Encoder

Firstly, we evaluate the multidimensional constellations learned by the encoder of Mul-VAE. 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 refers to one orthogonal resource. Fig. 5 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 each resource. More specifically, the encoder learns to transmit different messages with various power, as observed in different subfigures of Fig. 5, 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. Therefore, we can safely conclude that the proposed Mul-VAE learns to multiplex users by exploiting both the power and phase domains. Moreover, unlike SCMA where symbols are spanned sparsely, Mul-VAE allows each user to access all the resources, enhancing the robustness of the transmitting symbols and improving detection performance [55].

C. Evaluation of the Decoder

This part demonstrates the effectiveness of the attention operation, multi-task detection, and multi-task prioritization through training performance under both AWGN channel and Rayleigh fading channel. To show the training loss fluctuation clearly, we focus on the first 500 training losses and separate the low rate users' curves from high rate users.

1) Effect of Attention Operation: Fig. 6 depicts the training losses versus epochs with or without attention operation under AWGN channel and Rayleigh fading channel. It can be observed that the training losses are the same for either high rate users or low rate users under AWGN channel, whether the attention operation is employed or not. This is because there is no fading effect in AWGN channel, making the attention operation in (18) ineffectual. However, the attention operation effectively accelerates the convergence under Rayleigh fading channel, as shown in Fig. 6(c) and Fig. 6(d). This is because attention operation decompresses the received signal into a high dimension vector, which extracts adequate channel fading information and thus facilitates detection processes.

2) Effect of Multi-Task Detection: Fig. 7 compares the training losses with or without multi-task detection under AWGN channel and Rayleigh fading channel. Since the received signal is simple under AWGN channel, the decoder without multi-task detection is good enough to learn the MUD mapping. Thus introducing multi-task learning only slightly improves training losses, as observed in Fig. 7(a) and Fig. 7(b). However, multi-task detection becomes indispensable under Rayleigh fading channel, which significantly enhances the training losses and expedites the convergence, as illustrated in Fig. 7(c) and Fig. 7(d). This is because the transmitted symbols are greatly distorted by fading channels, and thus cannot

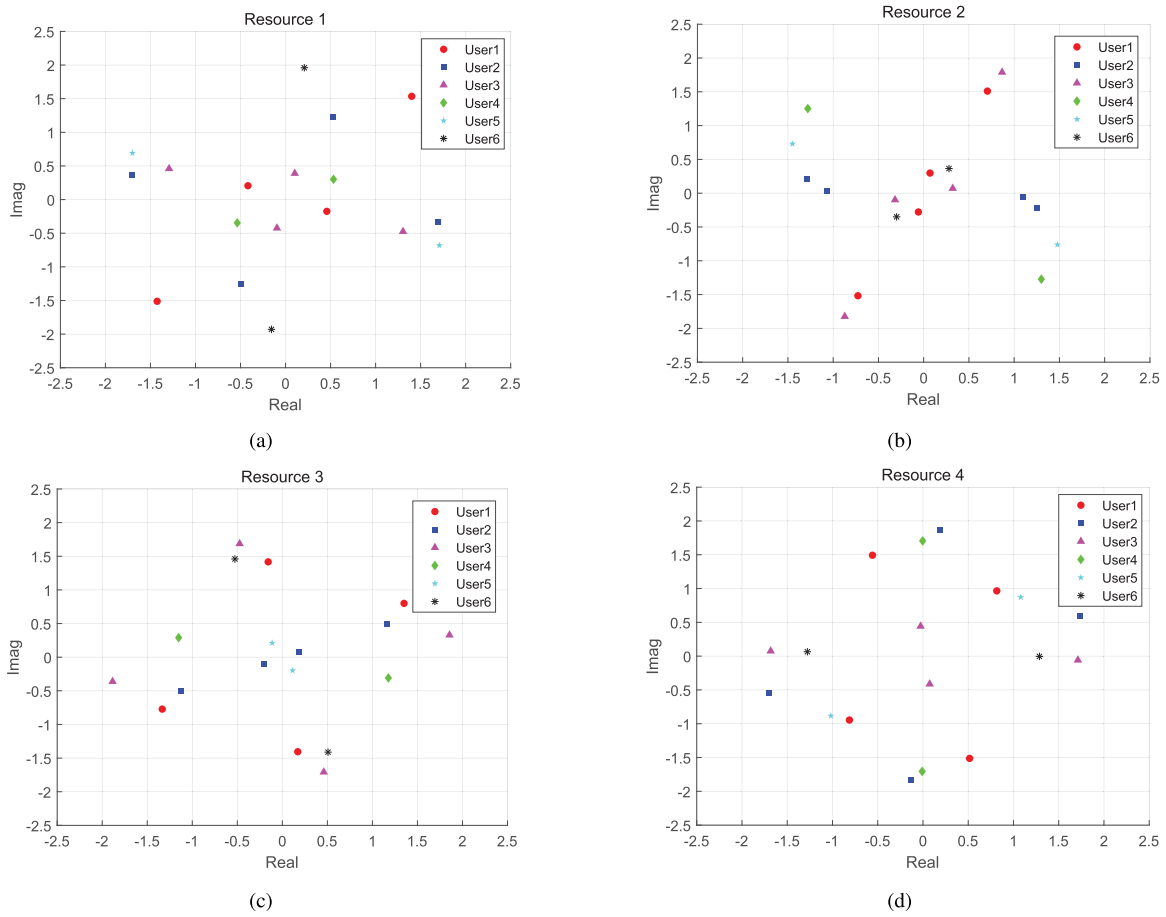


Fig. 5. The projection of the learned multidimensional constellations over 4 orthogonal resources.

be easily learned by the conventional decoder. Incorporating multi-task learning can exploit the correlation among different detection tasks, which improves the network learnability and enables the decoder to handle intricate signal structures.

3) Effect of Multi-Task Prioritization: Fig. 8 shows the training losses versus epochs with or without multi-task prioritization under AWGN channel and Rayleigh fading channel. As represented in Fig. 8(a) and Fig. 8(b), both the decoder with multi-task prioritization and the decoder without multi-task prioritization can achieve nearly the same training losses, thanks to the tractable signal structure in AWGN channel. It should be further noticed that although the decoder without multi-task prioritization has a faster convergence rate, the training loss gaps among different users are larger than those of the decoder with multi-task prioritization. However, the signal structure becomes complicated in Rayleigh fading channel, and all the users have to compete for limited computational resources. It can be clearly seen from Fig. 8(d) that some users are prioritized while the others are ignored if no prioritization strategy is adopted, which leads to colossal user unfairness and performance degradation. After deploying multi-task prioritization, we can see that both the training loss gaps of high rate users and low rate users are negligible. This is because the proposed multi-task prioritization adaptive adjusts the weights of different users through the focal loss, thereby enabling fair and efficient training by suitably allocating computational resources.

D. Evaluation of Overall Performance

We compare the block error rate (BLER) performance, i.e., $\mathbb{P}(\hat{s}_n \neq s_n)$, of Mul-VAE with competing methods. For clarity, we focus on the averaged BLERs of high rate users and low rate users. Fig. 9(a) depicts the BLERs of Mul-VAE and competing methods under AWGN channel. Notably, DL-based methods outperform the conventional methods by a large margin. For example, Mul-VAE and E2E achieve around 9.5 dB gain over SCMA-MPA and MUSA-SIC when SNR is higher than 0 dB, which demonstrates the superiority of joint optimization and shows the potential of end-to-end learning. Moreover, Mul-VAE 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. 9 compares the BLER of Mul-VAE with competing schemes under Rayleigh fading channel. We observe that Mul-VAE still achieves the lowest BLER among all methods, but the performance gaps between DL-based and conventional methods are narrowed. This is because the Rayleigh fading channel is randomly generated, and it is not easy for DNNs to suppress random fading effects through limited training samples and network size. Besides, the BLER of Mul-VAE

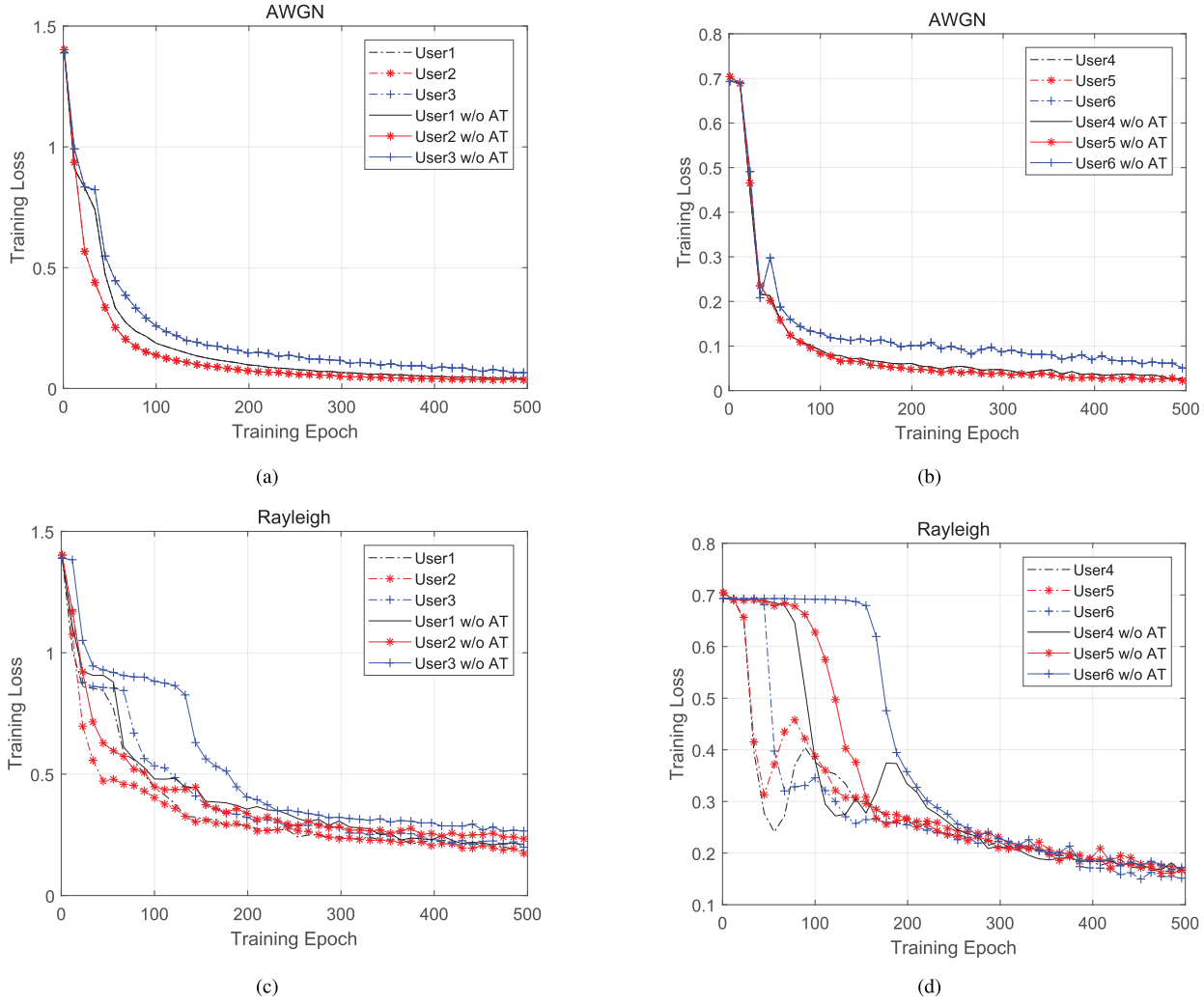


Fig. 6. Comparisons on the training losses of Mul-VAE with or without attention operation: (a) high rate users under AWGN channel, (b) low rate users under AWGN channel, (c) high rate users under Rayleigh fading channel, and (d) low rate users under Rayleigh fading channel.

keeps decreasing as SNR increases, while there are error floors in the BLER of the E2E method. The reason is as follows: (i) With no prioritization strategy employed, all users need to compete for computation resources, and some users are ignored while the others are prioritized, as shown in Fig. 8; (ii) The BLERs of ignored users fail to descend while the BLERs of prioritized users decrease sharply. Hence, ignored users will dominate the averaged BLERs at the high SNR region, as the BLERs of prioritized users are negligible. It can also be observed that the E2E method has a lower BLER than Mul-VAE for low rate users at the low SNR region, while Mul-VAE performs better for high rate users. This is because the E2E method puts more efforts into easy tasks while Mul-VAE concentrates more on challenging tasks, which also demonstrates the effectiveness of the proposed multi-task prioritization strategy.

E. Performance Fluctuation Over Different Initial Parameters

This part investigates the performance fluctuation of Mul-VAE over different initial parameters under both AWGN channel and Rayleigh fading channel. Three sets of parameter values are used for comparison, i.e., $\mu = -1\&\gamma = 2$, $\mu = -1\&\gamma = 4$, and $\mu = 1\&\gamma = 2$. Fig. 10(a) depicts the BLERs of Mul-VAE with different initial parameters under AWGN channel. It is seen that both Mul-VAE with $\mu = -1\&\gamma = 2$ and $\mu = 1\&\gamma = 2$ achieve the lowest BLER and outperform Mul-VAE with $\mu = -1\&\gamma = 4$ by about 1 dB when SNR is higher than 0 dB. The reason is as follows: (i) There is no fading effect in AWGN channel, so the attention operation in (18) is ineffectual. Therefore, the value of the product hyper-parameter μ does not affect the BLER performance of Mul-VAE. (ii) It is experimentally observed in [44] that $\gamma = 2$ leads to the best performance in the focal loss. Consequently, when μ remains the same, Mul-VAE with $\gamma = 2$ performs better than that with $\gamma = 4$.

Fig. 10 shows the BLERs of Mul-VAE with different initial parameters under Rayleigh fading channel. It can be observed that Mul-VAE with $\mu = -1\&\gamma = 2$ still yields the best performance, but the performance of Mul-VAE with $\mu = 1\&\gamma = 2$ is degraded under Rayleigh fading channel. This is because the attention operation with $\mu = -1$ can effectively decompress

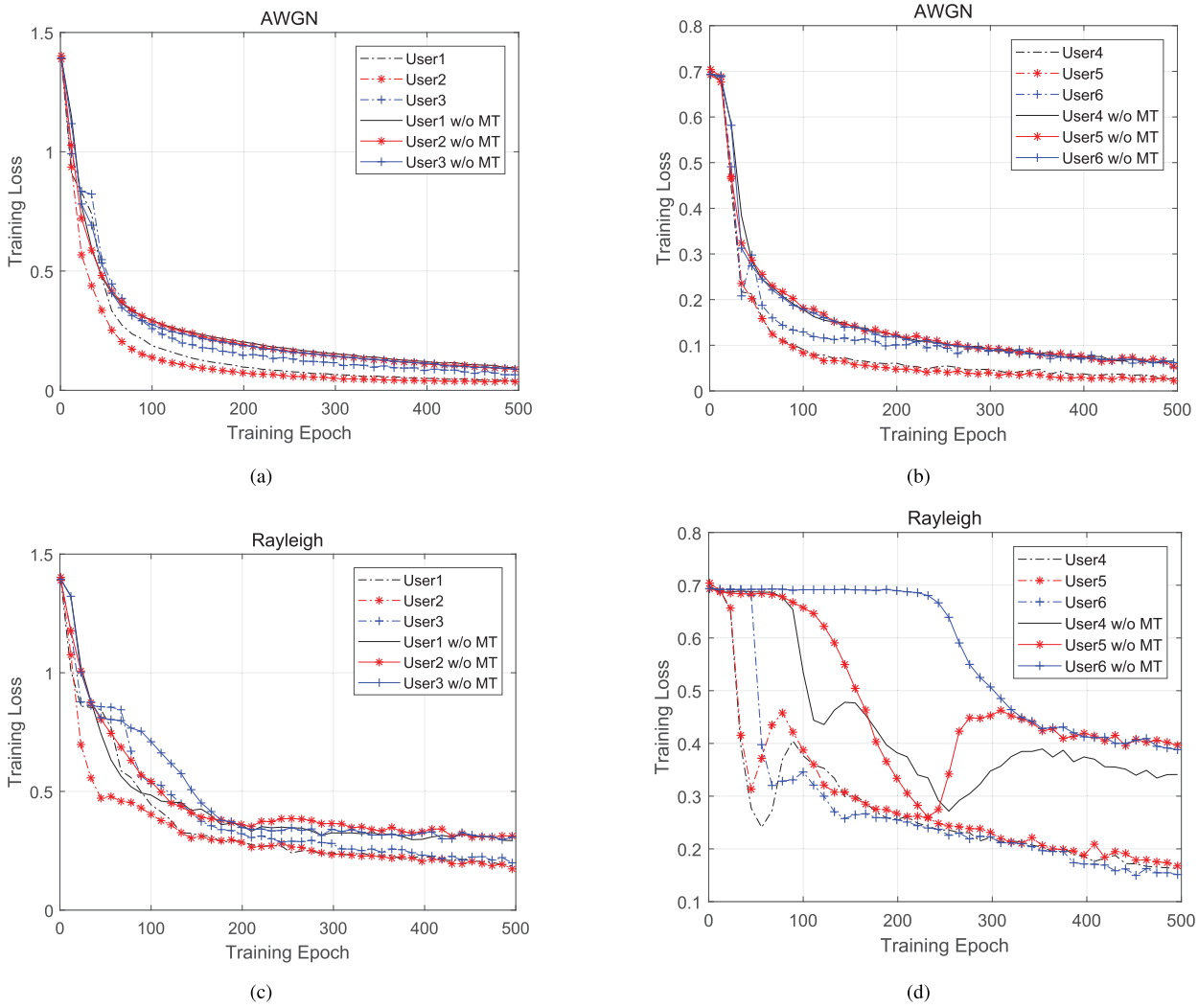


Fig. 7. Comparisons on the training losses of Mul-VAE with or without multi-task detection: (a) high rate users under AWGN channel, (b) low rate users under AWGN channel, (c) high rate users under Rayleigh fading channel, and (d) low rate users under Rayleigh fading channel.

the received signal into relatively independent signal streams as it is conceptually analogous to the zero-forcing equalization, while the attention operation with $\mu = 1$ cannot. Moreover, it can be noticed that Mul-VAE with $\mu = -1\&\gamma = 2$ slightly outperforms Mul-VAE with $\mu = -1\&\gamma = 4$ by around 0.1 dB, which is also consistent with the observations in [44].

F. Computational Complexity Analysis

In this part, we analyze the computational complexity of Mul-VAE and conventional methods, which is measured by the number of multiplication operations. For Mul-VAE, we focus on the computational complexity of online implementation. For conventional methods, we mainly consider the detection complexity, since the constellations can be designed in advance. The detailed complexity analysis is given as follows.

1) Mul-VAE: The complexity of DNN is mainly resulted from the multiplication between the weight matrix and the input vector, which can be seen as a whole bunch of dot products. For a layer with N_{l-1} input and N_l output, each dot product happens between the input vector and one column in $\mathbf{W}_l$ that counts as N_{l-1} multiplications. Since we have to compute N_l of these dot products, the total number of multiplications required by layer l is $N_{l-1}N_l$. Therefore, the complexity of one DNN with L layers can be expressed as

$$C_{DNN} = \sum_{l=1}^{L-1} N_l N_{l+1}. \tag{23}$$

As mentioned in Section IV-A, the encoder consists of N DNNs, each with an input layer (dimension of 2^R) followed by 4 hidden layers with $2^{(R+3)}$ neurons and an output layer with $2K$ dimension. Thus the total complexity of the encoder is

$$C_E = \sum_{n=1}^{N} (2^{2R_n} 200 + 16K2^{R_n}). \tag{24}$$

For the decoder, the complexity of the attention operation in the decoupling module is calculated as

$$C_A = 4K^2 N. \tag{25}$$

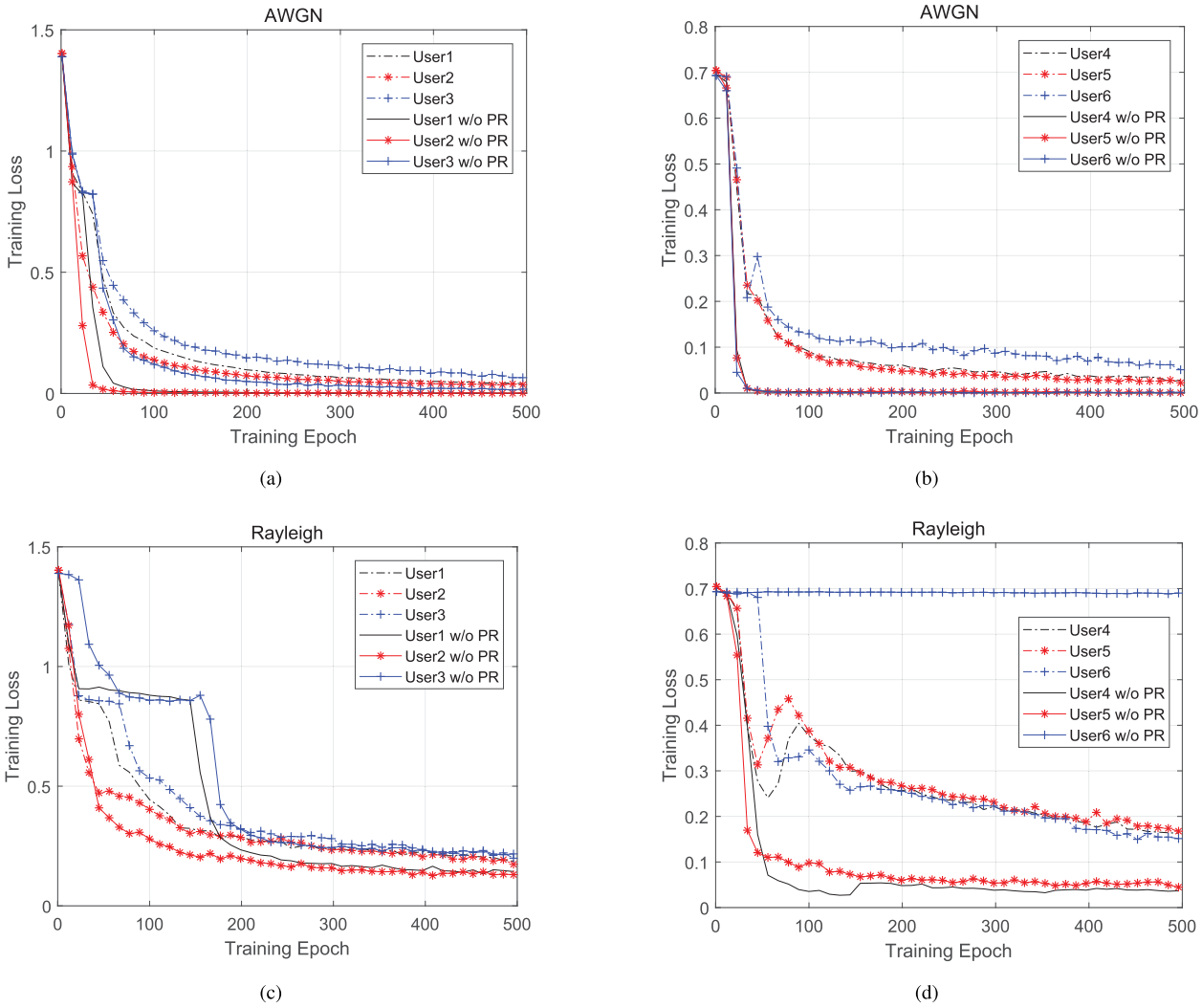


Fig. 8. Comparisons on the training losses of Mul-VAE with or without multi-task prioritization: (a) high rate users under AWGN channel, (b) low rate users under AWGN channel, (c) high rate users under Rayleigh fading channel, and (d) low rate users under Rayleigh fading channel.

The two-layer sluice network in the multi-task detection has $2N$ DNNs and 1 sluice, where the sluice comprises 1 DNN with $\sum_{n=1}^{N} 2^{R_n}$ inputs and N^2+N outputs. Since each DNN has 5 hidden layers with $2^{(R+3)}$ neurons, the complexity of multi-task detection is given by

$$\begin{aligned} C_M &= 2\sum_{n=1}^{N}(2^{2R_n}264 + 16K2^{R_n}) \\ &\quad + 8\sum_{n=1}^{N} 2^{2R_n} + 2^{2R_n}256 + 8(N^2+N)2^{R_n} \\ &= \sum_{n=1}^{N}(2^{2R_n}536 + 32K2^{R_n}) + 2^{2R_n}256 \\ &\quad + 8(N^2+N)2^{R_n}. \end{aligned} \tag{26}$$

From (24) to (26), the complexity of Mul-VAE is summarized as

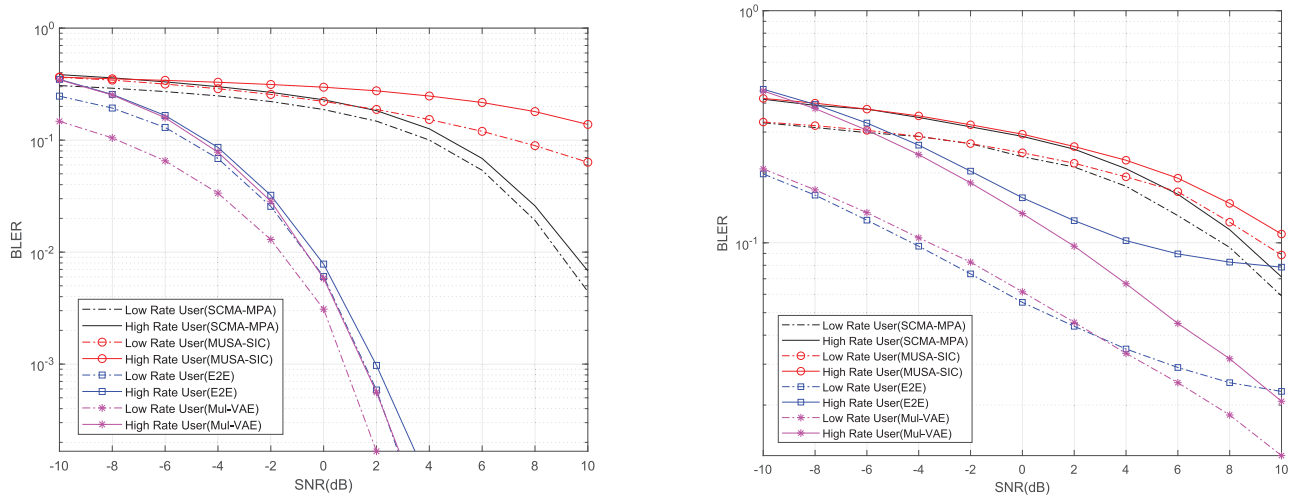
$$\begin{aligned} C_{Mul-VAE} &= \sum_{n=1}^{N}(2^{2R_n}736 + 48K2^{R_n}) \\ &\quad + 2^{2R_n}256 + 8(N^2+N)2^{R_n} + 4K^2N. \end{aligned} \tag{27}$$

2) MPA: The main complexity of MPA comes from the message passing from function nodes to variable nodes, i.e., equation (12.10) in [56]. The summation over all possible codewords adds up $\prod_{i\in\partial k\setminus j} 2^{R_i}$ terms and 2^{R_j} probabilities should be calculated in each iteration, where $\partial k \setminus j$ is the neighborhood of function node k excluding variable node j. The calculation in each summation can be divided into two parts, namely the exponent calculation for posterior probability and the product calculation for prior probability. Assuming multiplication has the same computational time with exponent operation, we have a total of 2 d_f multiplication operations in each summation, where d_f is the number of superimposed user on each resource. Since there are $d_f K$ message passing paths, the complexity of the MPA detector is given by

$$C_{MPA} = 2N_{iter}Kd_f^2\prod_{i=1}^{d_f} 2^{R_i}. \tag{28}$$

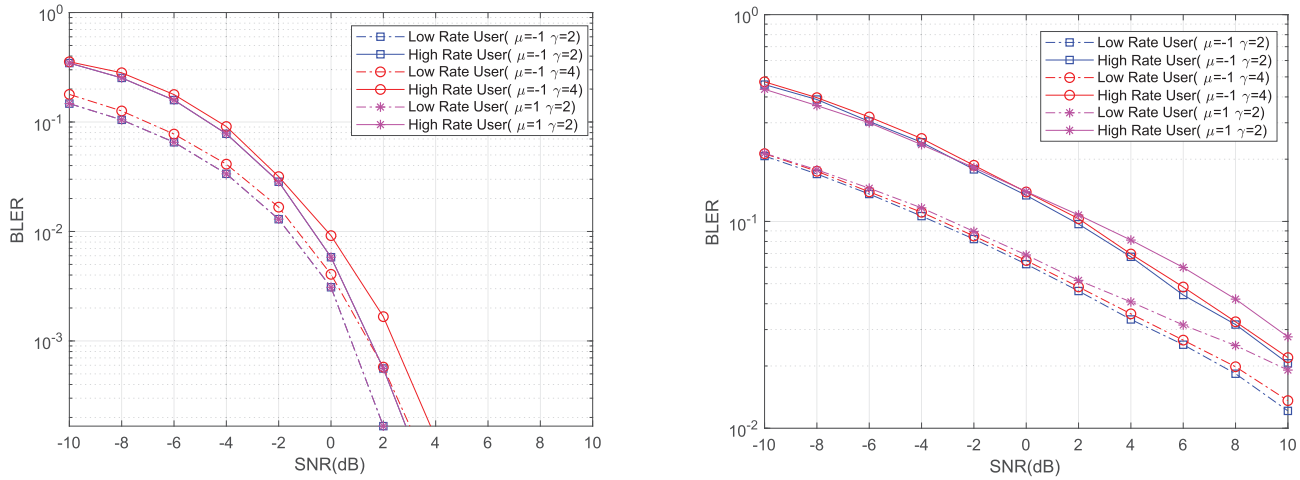
According to [9], we take $d_f = 2N/K$ and simplify (28) as

$$C_{MPA} = \frac{8N_{iter}N^2}{K}\prod_{i=1}^{2N/K} 2^{R_i}. \tag{29}$$



(a) (b)

Fig. 9. BLER performance comparisons under (a) AWGN channel and (b) Rayleigh fading channel.



(a) (b)

Fig. 10. BLER performance with different initial parameters under (a) AWGN channel and (b) Rayleigh fading channel.

TABLE I

COMPUTATIONAL COMPLEXITY COMPARISON ($K = 4$, $R = 1$)

	Complexity expression	Complexity with various user numbers		
		$N = 50$	$N = 100$	$N = 500$
Mul-VAE	$\sum_{n=1}^{N}(2^{2R_n}736 + 48K2^{R_n}) + 2^{2R_n}256 + 8(N^2+N)2^{R_n} + 4K^2N$	2.11×10^5	5.02×10^5	5.71×10^6
MPA	$\frac{8N_{iter}N^2}{K}\prod_{i=1}^{2N/K} 2^{R_i}$	8.39×10^{11}	1.13×10^{20}	4.52×10^{81}
MMSE-SIC	$2N^3K + 4N^2K + 2NK$	1.04×10^6	8.16×10^6	1.01×10^9

3) MMSE-SIC: Since the complexity of MMSE-SIC detector has been analyzed in the literature [57], we omit the details for brevity and express the complexity as

$$C_{MMSE-SIC} = 2N^3K + 4N^2K + 2NK. \tag{30}$$

We summarize the complexity of the above algorithms in Table I. To examine the behavior with various user numbers, we compute the complexity for $N = 50$, 100, and 500, with K set as 4 and R set as 1 bit/s. We observe that the complexity of Mul-VAE is much smaller than that of conventional methods. For example, when $N = 50$, the complexity of Mul-VAE is only $2.5 \times 10^{-5}\%$ and 20.3% of those of MPA and MMSE-SIC, respectively. Moreover, when N increases, the complexity of Mul-VAE increases marginally while those of MPA and MMSE-SIC maintain a significant upward trend. These observations show the superiority of

Mul-VAE in terms of computational complexity, indicating the capability of Mul-VAE to support massive connectivity with low processing delay.

V. Conclusion

This paper has proposed a novel DL-based method for the joint design of MCD and MUD in the GF-NOMA system, where a variational autoencoder based-network is developed to approximate the optimal MCD and MUD in a probabilistic and holistic manner. Based on the insights from theoretical analysis on the loss function, we design a multi-task learning structure to tackle the mutually conflicting yet related MUD processes. We also provide a multi-task prioritizing strategy to alleviate the training unfairness among users. Extensive simulation results validate that the proposed scheme can significantly improve the detection accuracy with low computational complexity. In this paper, we restrict our studies to the joint optimization for GF-NOMA, but we believe that there are many interesting applications of the proposed Mul-VAE, such as precoding and decoding in MIMO and multimodal fusion. For the future work, we will investigate the implementation of Mul-VAE in mobile devices, where model pruning and knowledge distilling can be utilized to reduce the network complexity. Moreover, meta-learning and dynamic neural networks can also be considered to improve the adaptiveness of Mul-VAE to changing environments.

Appendix A
Proof of Theorem 1

The proof goes as follows:

$$
\begin{aligned}
&\log P(\mathbf{s};[g_n]_{n=1}^N)\\
&= \int Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)\log P(\mathbf{s};[g_n]_{n=1}^N)\,d\mathbf{y}\\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{P(\mathbf{s},\mathbf{y};[g_n]_{n=1}^N)Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}{P(\mathbf{y}|\mathbf{s};[g_n]_{n=1}^N)Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\right]\\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{P(\mathbf{s},\mathbf{y};[g_n]_{n=1}^N)}{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\right]\\
&\quad+\mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}{P(\mathbf{y}|\mathbf{s};[g_n]_{n=1}^N)}\right]\\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{P(\mathbf{s}|\mathbf{y};[g_n]_{n=1}^N)P(\mathbf{y})}{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\right]\\
&\quad+KL(Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)||P(\mathbf{y}|\mathbf{s};[g_n]_{n=1}^N))\\
&\overset{(a)}{\geq} \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{P(\mathbf{s}|\mathbf{y};[g_n]_{n=1}^N)P(\mathbf{y})}{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\right]\\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}[\log P(\mathbf{s}|\mathbf{y};[g_n]_{n=1}^N)]\\
&\quad-\mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}\left[\log\frac{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}{P(\mathbf{y})}\right]\\
&= \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}[\log P(\mathbf{s}|\mathbf{y};[g_n]_{n=1}^N)]\\
&\quad-KL(Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)||P(\mathbf{y}))\\
&\overset{(b)}{=} \sum_{n=1}^N \mathbb{E}_{Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)}[\log P(s_n|\mathbf{y};g_n)]\\
&\quad-KL(Q(\mathbf{y}|\mathbf{s};[f_n]_{n=1}^N)||P(\mathbf{y})),
\end{aligned}
\tag{31}
$$

where (a) holds due to the non-negativity of the Kullback-Leibler divergence [58], and (b) holds since s_n is independent with each other.

Appendix B
Proof of Theorem 2

Denoting the correct classification probability as P_C, the cross entropy loss with one-hot encoding can be expressed as

$$\mathcal{L}_C = -\log P_C. \tag{32}$$

Considering a batch with B samples, the average cross entropy loss over the whole batch is

$$\mathcal{L}_C = -\frac{1}{B}\sum_{i=1}^{B}\log P_C^{(i)}. \tag{33}$$

According to the derivation in [59], the AM-GM ratio, i.e., $\frac{(\prod_{i=1}^B P_C^{(i)})^{1/B}}{\frac{1}{B}\sum_{i=1}^B P_C^{(i)}}$, is almost constant as $B\to\infty$. That is, $\lim\limits_{B\to\infty}\frac{(\prod_{i=1}^B P_C^{(i)})^{1/B}}{\frac{1}{B}\sum_{i=1}^B P_C^{(i)}} = c$, where $c\in(0,1]$ is the constant of proportionality. Therefore, when the batch size goes to infinity, we can obtain

$$
\begin{aligned}
&\lim_{B\to\infty}\mathcal{L}_C\\
&= \lim_{B\to\infty} -\frac{1}{B}\sum_{i=1}^{B}\log P_C^{(i)} = \lim_{B\to\infty} -\log(\prod_{i=1}^{B}P_C^{(i)})^{\frac{1}{B}}\\
&= \lim_{B\to\infty} -\log\frac{c}{B}\sum_{i=1}^{B}P_C^{(i)} \approx -\log\mathbb{E}[cP_C] = -\log c\kappa,
\end{aligned}
\tag{34}
$$

where $\kappa=\mathbb{E}[P_C]$ is the classification accuracy of the whole batch, and c can be experimentally determined as 0.7357 via extensive simulations. Accordingly, the relationship between $\mathcal{L}_C$ and κ can be approximated by

$$\kappa \approx \frac{e^{-\mathcal{L}_C}}{0.7357}. \tag{35}$$

References

[1] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1805–1838, 1st Quart., 2020.

[2] Y. Dai, M. Sheng, J. Liu, N. Cheng, X. Shen, and Q. Yang, "Joint mode selection and resource allocation for D2D-enabled NOMA cellular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6721–6733, Jul. 2019.

[3] L. Lyu, C. Chen, N. Cheng, S. Zhu, X. Guan, and X. Shen, "NOMA-assisted on-demand transmissions for monitoring applications in industrial IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12264–12276, Oct. 2020.

[4] Y. Wang, Z. Tian, and X. Cheng, "Enabling technologies for spectrum and energy efficient NOMA-MmWave-MaMIMO systems," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 53–59, Oct. 2020.

[5] G. Yang, C. K. Ho, and Y. L. Guan, "Multi-antenna wireless energy transfer for backscatter communication systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2974–2987, Dec. 2015.

[6] G. Yang, Y.-C. Liang, R. Zhang, and Y. Pei, "Modulation in the air: Backscatter communication over ambient OFDM carrier," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1219–1233, Mar. 2018.

[7] G. Yang, Q. Zhang, and Y.-C. Liang, "Cooperative ambient backscatter communications for green Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1116–1130, Apr. 2018.

[8] C. H. Liu and D. C. Liang, "Heterogeneous networks with powerdomain NOMA: Coverage, throughput, and power allocation analysis," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3524–3539, May 2018.

[9] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Proc. IEEE 24th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2013, pp. 332–336.

[10] Z. Yuan, G. Yu, W. Li, Y. Yuan, X. Wang, and J. Xu, "Multi-user shared access for Internet of Things," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[11] 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. 10543–10553, Nov. 2018.

[12] K. Xiao, B. Xia, Z. Chen, B. Xiao, D. Chen, and S. Ma, "On capacity-based codebook design and advanced decoding for sparse code multiple access systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3834–3849, Jun. 2018.

[13] Y. Zhou, Q. Yu, W. Meng, and C. Li, "SCMA codebook design based on constellation rotation," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[14] S. Liu, J. Wang, J. Bao, and C. Liu, "Optimized SCMA codebook design by QAM constellation segmentation with maximized MED," *IEEE Access*, vol. 6, pp. 63232–63242, 2018.

[15] 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. 2019.

[16] J. Dai, K. Niu, C. Dong, and J. Lin, "Improved message passing algorithms for sparse code multiple access," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9986–9999, Nov. 2017.

[17] L. Yuan, J. Pan, N. Yang, Z. Ding, and J. Yuan, "Successive interference cancellation for LDPC coded nonorthogonal multiple access systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5460–5464, Jun. 2018.

[18] 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.

[19] C. Wei, H. Liu, Z. Zhang, J. Dang, and L. Wu, "Approximate message passing-based joint user activity and data detection for NOMA," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 640–643, Mar. 2017.

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

[21] 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, Feb. 2020.

[22] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[23] A. Zappone, M. D. Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[24] W. Wu *et al.*, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, Jul. 2021.

[25] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, 2020.

[26] W. Wu, N. Cheng, N. Zhang, P. Yang, W. Zhuang, and X. Shen, "Fast mmWave beam alignment via correlated bandit learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 12, pp. 5894–5908, Dec. 2019.

[27] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[28] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018.

[29] F. Alberge, "Deep learning constellation design for the AWGN channel with additive radar interference," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1413–1423, Feb. 2019.

[30] 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.

[31] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[32] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication PHY using neural network," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1364–1373, Jun. 2019.

[33] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.

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

[35] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*. [Online]. Available: http://arxiv.org/abs/1606.05908

[36] R. Wei, C. Garcia, A. El-Sayed, V. Peterson, and A. Mahmood, "Variations in variational autoencoders—A comparative evaluation," *IEEE Access*, vol. 8, pp. 153651–153670, 2020.

[37] B. Esmaeili *et al.*, "Structured disentangled representations," in *Proc. AISTATS*, 2019, pp. 2525–2534.

[38] I. Higgins *et al.*, "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. ICLR*, 2017, pp. 1–13.

[39] T. Zhao, F. Li, and P. Tian, "A deep-learning method for device activity detection in mMTC under imperfect CSI based on variational-autoencoder," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7981–7986, Jul. 2020.

[40] A. Caciularu and D. Burshtein, "Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 3, pp. 1003–1018, Sep. 2020.

[41] M. Guo, A. Haque, D. Huang, S. Yeung, and F. Li, "Dynamic task prioritization for dynamic task learning," in *Proc. ECCV*, 2018, pp. 282–299.

[42] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.

[43] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proc. AAAI*, 2019, pp. 4822–4829.

[44] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for denseobject detection," in *Proc. IEEE ICCV*, 2017, pp. 2999–3007.

[45] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: http://arxiv.org/abs/1706.05098

[46] N. Slamnik-Krijestorac, H. Kremo, M. Ruffini, and J. M. Marquez-Barja, "Sharing distributed and heterogeneous resources toward end-to-end 5G networks: A comprehensive survey and a taxonomy," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1592–1628, 3rd Quart., 2020.

[47] X. Yang, "Understanding the variational lower bound," Tech. Rep., 2017. [Online]. Available: https://xyang35.github.io/2017/04/14/variational-lower-bound/

[48] Y. Du *et al.*, "Joint channel estimation and multiuser detection for uplink grant-free NOMA," *IEEE Commun. Lett.*, vol. 7, no. 4, pp. 682–685, Aug. 2018.

[49] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Joint channel coding and modulation via deep learning," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.

[50] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[51] M. Stark, F. A. Aoudia, and J. Hoydis, "Joint learning of geometric and probabilistic constellation shaping," in *Proc. IEEE GLOBECOM*, Dec. 2019, pp. 1–5.

[52] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[53] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," 2018, *arXiv:1804.07612*. [Online]. Available: http://arxiv.org/abs/1804.07612

[54] *Altera Innovate Asia 1st 5G Algorithm Innovation Competition*. Accessed: 2015. [Online]. Available: http://www.innovateasia.com/5g/gp2.html

[55] M. Vameghestahbanati, I. D. Marsland, R. H. Gohary, and H. Yanikomeroglu, "Multidimensional constellations for uplink SCMA systems—A comparative study," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2169–2194, 3rd Quart., 2019.

[56] M. Vaezi, Z. Ding, and H. V. Poor, *Multiple Access Techniques for 5G Wireless Networks and Beyond*. Berlin, Germany: Springer, 2019.

[57] J.-H. Park, Y. Whang, and K. S. Kim, "Low complexity MMSE-SIC equalizer employing time-domain recursion for OFDM systems," *IEEE Signal Process. Lett.*, vol. 15, no. 3, pp. 633–636, Oct. 2008.

[58] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[59] J. M. Aldaz, "Concentration of the ratio between the geometric and arithmetic means," *J. Theor. Probab.*, vol. 23, no. 2, pp. 498–508, Jun. 2010.

Zhe Ma (Student Member, IEEE) received the B.E. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Department of Automation, Tsinghua University, Beijing. He was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Waterloo, in 2020. His research interests include the Internet of Things, massive random access, and deep learning.

Wen Wu (Member, IEEE) received the B.E. degree in information engineering from South China University of Technology, Guangzhou, China, the M.E. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2012 and 2015, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019. Since 2019, he works as a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include millimeter-wave networks and AI-empowered wireless networks.

Mengnan Jian received the B.E. degree in information engineering from Beijing Institute of Technology, Beijing, China, in 2016, and the M.S. degree from Tsinghua University, Beijing, in 2019. She is currently an Engineer with ZTE Corporation. Her research interests include orbital angular momentum and reconfigurable intelligent surface.

Feifei Gao (Fellow, IEEE) received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2002, the M.Sc. degree from McMaster University, Hamilton, ON, Canada, in 2004, and the Ph.D. degree from the National University of Singapore, Singapore, in 2007.

In 2011, he joined the Department of Automation, Tsinghua University, Beijing, China, where he is currently an Associate Professor. He has authored/coauthored more than 150 refereed IEEE journal articles and more than 150 IEEE conference proceeding papers that are cited more than 11 000 times in Google Scholar. His research interests include signal processing for communications, array signal processing, convex optimizations, and artificial intelligence assisted communications. He has served as an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (Lead Guest Editor), IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE SIGNAL PROCESSING LETTERS (Senior Editor), IEEE COMMUNICATIONS LETTERS (Senior Editor), IEEE WIRELESS COMMUNICATIONS LETTERS, and *China Communications*. He has also served as the Symposium Co-Chair for 2019 IEEE Conference on Communications (ICC), 2018 IEEE Vehicular Technology Conference Spring (VTC), 2015 IEEE Conference on Communications (ICC), 2014 IEEE Global Communications Conference (GLOBECOM), and 2014 IEEE Vehicular Technology Conference Fall (VTC), as well as a Technical Committee Member of more than 50 IEEE conferences.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular networks. He is a registered Professional Engineer in ON, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (ON) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award from the University of Waterloo in 2006 and the Premiers Research Excellence Award (PREA) in 2003 from ON. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President Elect of the IEEE ComSoc. He was the Vice President of Technical and Educational Activities, the Vice President of Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, *IEEE Network*, and *IET Communications*.