This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3087132, IEEE Transactions on Vehicular Technology

Multi-Label Learning Based Antenna Selection in Massive MIMO Systems

Wentao Yu, Tianyu Wang, *Member, IEEE*, and Shaowei Wang, *Senior Member, IEEE*

***Abstract*—Antenna selection (AS) is a signal processing technology that can greatly reduce the hardware complexity of multi-antenna systems. Specifically, AS can decrease the number of required radio frequency chains by activating only a subset of the available antennas in each transmission slot. However, optimal AS suffers from a high computational complexity that increases exponentially with the scale of the antenna array. In this paper, we propose a low-complexity AS algorithm based on multi-label learning (MLL), where a deep neural network is employed to determine the set of selected antennas for a given channel matrix. Specifically, the MLL network combines deep canonical correlation analysis and an autoencoder in a unified network structure, which can extract the low dimensional features of channel matrix as well as the interdependency among selected antennas, so as to achieve an accurate prediction of the set of selected antennas with a relatively small-scale learning model. Simulation results show that, in comparison with the convex relaxation based method, our proposed MLL-based method can achieve comparable capacity with significantly reduced computation time.**

***Index Terms*—Antenna selection, deep neural network, massive MIMO, multi-label learning.**

I. Introduction

MASSIVE MIMO has been proven a promising technique to offer enormous improvements in spectral and energy efficiency by using a large-scale antenna array [1]. However, the number of radio frequency (RF) chains of a massive MIMO system is required to be equal to the number of antennas such that the spatial degrees of freedom can be fully exploited. This hardware requirement brings great economic and technical challenges to the design of wireless transceivers. To alleviate this problem, antenna selection (AS) has been proposed, where the number of RF chains can be greatly reduced by activating only a subset of available antennas. Since different antennas usually make unequal contributions in time-varying propagation environments, AS is performed periodically in each transmission time interval (TTI).

AS has been extensively studied in conventional MIMO systems, where the number of antennas is usually no more than 8 [2]. The optimal set of selected antennas can be calculated by exhaustive search in conventional MIMO, however, it is computationally prohibitive owing to the underlying combinatorial complexity in massive MIMO cases [3]. The formulated optimization task is NP-hard and can be transformed into a convex optimization problem by relaxing the integer constraints [4]. Genetic algorithms can also be applied to directly solve the AS problem as discussed in [5, 6]. Due to the large-scale antenna array, these algorithms require a considerable number of iterative calculations, which is still computationally unacceptable in practice. Low-complexity AS algorithms for massive MIMO are preferred. A greedy algorithm that can achieve a constant-factor worst case performance guarantee is proposed in [7] by leveraging submodularity. Matching pursuit technique are utilized to minimize the mean squared reception error [8]. Power-based algorithms can greatly simplify the calculation by merely using the received signal power of each antenna [4, 6]. Two low-complexity algorithms that can avoid vector multiplications during the AS procedure are presented for multi-user massive MIMO systems with matched filter precoding [9]. However, low computational complexity usually comes at a cost of considerable performance degradation.

To achieve a tradeoff between the computational complexity and the system performance, machine learning based methods have been proposed recently [10–14]. Most of these works focus on AS in conventional MIMO systems. In [10, 11], the authors respectively formulate AS in single-user and multi-user MIMO systems as a series of binary classification problems and employs two classical machine learning algorithms, i.e., support vector machine and k-nearest neighbours, to determine the selected antennas. However, these methods require a large number of binary classifiers, which results in high computational complexity. In addition, the interdependency between selected antennas in a large-scale antenna array is difficult to be captured by using independent classifiers, which may lead to a considerable performance loss. In [12, 13], the authors utilize a multi-layer perceptron to learn the direct mapping from the channel matrix to the optimal set of selected antennas. However, the scales of such neural networks are directly determined by the underlying combinatorial complexity of the AS problem, which becomes unacceptable in massive MIMO. Recently, there are also some attempts of applying deep learning methods to AS in massive MIMO systems. In [14], the authors propose a pair of twin convolutional neural networks (CNNs), which are trained with a large amount of labeled data, to sequentially perform receive antenna selection and hybrid beamformer design in a single-user (massive) MIMO system, which differs from our system configuration. The convolution layers are often utilized to capture the relationship among nearby pixels in image processing tasks. However, the physical meaning of applying convolution over wireless channels is uncertain.

Manuscript received October 22, 2020; revised February 19, 2021 and May 7, 2021; accepted June 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 61801208, 61931023, and U1936202. The review of this article was coordinated by Prof. Jingon Joung. *(Wentao Yu and Tianyu Wang are co-first authors.)* *(Corresponding author: Shaowei Wang.)*

The authors are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China (e-mail: wentaoyu@smail.nju.edu.cn; tianyu.alex.wang@nju.edu.cn;wangsw@nju.edu.cn).

In this paper, we consider AS for the downlink of a multi-user massive MIMO system, for which a multi-label learning (MLL) based method is proposed. Specifically, the proposed method employs a deep neural network that combines deep canonical correlation analysis (DCCA) [15] and an autoencoder to simultaneously label each antenna according to the input channel matrix. Each label is binary and indicates whether the corresponding antenna should be selected or not. Different from [10–14], MLL can represent the relationship between the channel matrix and the corresponding selected antennas in a low-dimensional space, and can utilize the interdependency among selected antennas to further improve the model accuracy. Therefore, the size of the neural network employed by the MLL can be kept small as the scale of antenna array increases. Simulation results show that, in a typical massive MIMO system, the proposed MLL-based AS method can achieve a comparable capacity relative to the prevailing convex relaxation based one, while the computation time can be greatly reduced. Comparison with the CNN-based method shows that the proposed MLL-based method has advantages in both performance and computational complexity.

Notations: Vectors and matrices are respectively denoted by the lower-case and upper-case boldface letters; $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$, $\mathrm{tr}(\cdot)$ are the transpose, Hermitian transpose, reverse and trace of a matrix, respectively; $\|\cdot\|_0$ and $\|\cdot\|_F$ respectively stand for the ℓ_0-norm and the Frobenius norm; $\mathrm{diag}(\cdot)$ returns a square diagonal matrix with the entries of the input vector on its main diagonal; $\boldsymbol{I}$ denotes the identity matrix.

II. System Model

Consider the downlink transmission of a massive MIMO system, where the base station is equipped with K RF chains and N available antennas, while M single-antenna users receive data from the base station. Without loss of generality, we assume $N > K \gg M$. We denote by $\boldsymbol{H} \in \mathbb{C}^{M\times N}$ the overall channel matrix between the base station and the users.

In each TTI, the AS scheme needs to select a K-out-of-N subset of available antennas that are connected to the RF chains. We introduce an $N\times 1$ vector $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_N]^T$ to denote the set of selected antennas, where

$$\delta_i = \begin{cases} 1, & \text{if antenna } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The channel matrix of the selected antennas is then obtained by extracting the non-zero columns of $\boldsymbol{H}\,\mathrm{diag}(\boldsymbol{\delta})$. Normalization is performed thereafter so that each element has unit energy, averagely distributed over K antennas and M users. The normalized channel matrix is denoted by $\boldsymbol{H}^{(\boldsymbol{\delta})} \in \mathbb{C}^{M\times K}$. Thus, for any transmit signal $\boldsymbol{s} \in \mathbb{C}^M$, the received signal is given by

$$\boldsymbol{y} = \sqrt{\rho M}\boldsymbol{H}^{(\boldsymbol{\delta})}\boldsymbol{W}\boldsymbol{s} + \boldsymbol{n}, \tag{2}$$

where ρ stands for the average interference-free transmit SNR per user, ρM is the total transmit power, $\boldsymbol{W} \in \mathbb{C}^{K\times M}$ is the precoding matrix, and $\boldsymbol{n} \in \mathbb{C}^M$ is the noise vector satisfying i.i.d. complex Gaussian distribution with zero mean and unit variance. The performance of AS is evaluated under two widely used linear precoding schemes, i.e., zero-forcing (ZF) precoding and minimum-mean-square-error (MMSE) precoding, which are respectively given by [16]

$$\boldsymbol{W}_{ZF} = \left(\boldsymbol{H}^{(\boldsymbol{\delta})}\right)^H \left(\boldsymbol{H}^{(\boldsymbol{\delta})}\left(\boldsymbol{H}^{(\boldsymbol{\delta})}\right)^H\right)^{-1}, \tag{3}$$

$$\boldsymbol{W}_{MMSE} = \left(\boldsymbol{H}^{(\boldsymbol{\delta})}\right)^H \left(\boldsymbol{H}^{(\boldsymbol{\delta})}\left(\boldsymbol{H}^{(\boldsymbol{\delta})}\right)^H + \frac{1}{\rho K}\boldsymbol{I}\right)^{-1}. \tag{4}$$

Thus, the received SINR of user m is given by

$$SINR_m^{(\boldsymbol{\delta})} = \frac{\rho M \left|\boldsymbol{h}_m^{(\boldsymbol{\delta})}\boldsymbol{w}_m^{(\boldsymbol{\delta})}\right|^2}{1 + \rho M \sum_{m'\neq m}^{M} \left|\boldsymbol{h}_{m'}^{(\boldsymbol{\delta})}\boldsymbol{w}_{m'}^{(\boldsymbol{\delta})}\right|^2}, \tag{5}$$

where $\boldsymbol{h}_m^{(\boldsymbol{\delta})}$ represents the channel vector between the selected antennas and user m, i.e., the m-th row of $\boldsymbol{H}^{(\boldsymbol{\delta})}$, and $\boldsymbol{w}_m^{(\boldsymbol{\delta})}$ is the m-th column of the precoding matrix $\boldsymbol{W}$. The total downlink capacity is then given by

$$C = \sum_{m=1}^{M} \log_2\left(1 + SINR_m^{(\boldsymbol{\delta})}\right). \tag{6}$$

Denote $\mathcal{T} = \{\boldsymbol{\delta} \in \mathbb{R}^N \,|\, \delta_i \in \{0,1\}, \|\boldsymbol{\delta}\|_0 = K\}$ as the set of every possible AS vector. The optimal AS decision is then given by

$$\boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\delta}\in\mathcal{T}} C. \tag{7}$$

Note that (7) is an integer programming problem that is NP-hard. Although relaxation and heuristic methods can be applied [4–6], they usually come with the disadvantage of high computational complexity, especially for massive MIMO systems with a large number of antennas. We propose an MLL-based method to produce promising solutions efficiently.

III. MLL-Based Antenna Selection

We first formulate AS as an MLL problem. Then, we specify the neural network architecture, define a problem dependent loss function, and discuss the implementation details. At last, we briefly analyze the computational complexity of the related algorithms.

A. Antenna Selection as MLL

MLL refers to a type of classification problem where each input instance is simultaneously associated with multiple labels [17]. For example, in image recognition, a picture could cover several scenes simultaneously, such as sea, beach and coconut trees, while in text mining, an article could be associated with several topics such as urban, buildings and crowds. These labels usually exhibit high interdependency among each other, which makes MLL fundamentally different from the conventional single-label classification tasks. In the considered AS problem, the full channel matrix $\boldsymbol{H}$ can be treated as the input instance that arrives every TTI. The AS vector $\boldsymbol{\delta}$ is treated as the label vector with N interdependent labels. Thus, AS in massive MIMO systems can be transformed into an MLL problem that aims to establish a classifier to predict $\boldsymbol{\delta}$ from $\boldsymbol{H}$.

Although MLL can be intuitively decomposed into a series of independent single-label classification problems, it has

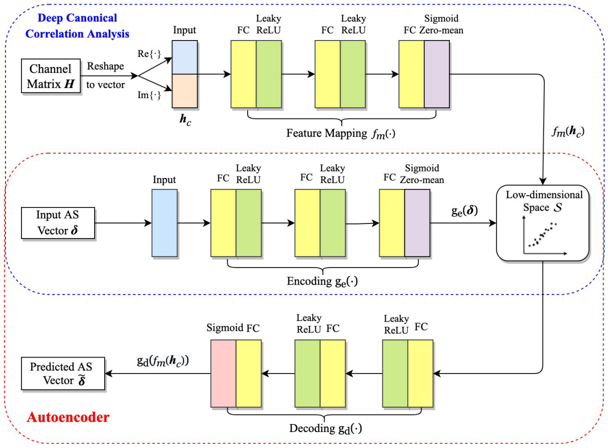

Fig. 1. Multi-label learning for antenna selection in massive MIMO.

been shown that such a decomposition ignores the correlation among labels and fails to extract the underlying combinatorial complexity when the number of labels is large [17]. Hence, conventional single-label learning based AS methods can only cope with a small number of antennas [10]. Here, we introduce a multi-label neural network that can extract the correlation among antenna labels as well as the low-dimensional features of the channel matrix, which enables a relatively small-sized neural network to accurately predict the AS vector for a large-scale antenna array. Moreover, the multi-label neural network can determine the labels of all available antennas in one shot, which greatly reduces the computation time as compared to the conventional method with a sequence of single-label classifiers. Lastly, the online computation of the trained MLL network can be highly parallelized and accelerated by using graphics processing units (GPUs) so that the computation burden can be further decreased.

B. Network Architecture

As shown in Fig. 1, the MLL network consists of three components: encoding network $g_e(\cdot)$, decoding network $g_d(\cdot)$ and feature mapping network $f_m(\cdot)$. They all comprise three fully connected (FC) layers with dedicated activation functions [18], including leaky ReLU, sigmoid, and sigmoid zero-mean (i.e., $\text{sigmoid}(\cdot) - 0.5$), which are given by

$$\text{leaky ReLU}(z) = \begin{cases} z & \text{if } z > 0, \\ 0.01z & \text{if } z \le 0, \end{cases} \tag{8}$$

and

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}. \tag{9}$$

The input of $f_m(\cdot)$, denoted by $\boldsymbol{h}_c$, is a $2MN \times 1$ real vector obtained by partitioning and reshaping the channel matrix $\boldsymbol{H}$. The output is an $S \times 1$ vector $f_m(\boldsymbol{h}_c)$ in a low-dimensional space $\mathcal{S}$. The input of $g_e(\cdot)$ is the optimal AS vector $\boldsymbol{\delta}$ for the input channel matrix $\boldsymbol{H}$, and the output is an $S \times 1$ vector $g_e(\boldsymbol{\delta})$ in $\mathcal{S}$. The input of $g_d(\cdot)$ is the output of $f_m(\cdot)$, given by $f_m(\boldsymbol{h}_c)$, and the output of it is an $N \times 1$ vector given by $\tilde{\boldsymbol{\delta}} = g_d(f_m(\boldsymbol{h}_c))$, which represents the predicted AS vector. These component networks constitute two distinct function blocks, i.e., the DCCA and the autoencoder, which are respectively outlined by dotted lines in Fig. 1.

DCCA is able to learn a pair of deep neural network based projections to transform two views of related data into a common low-dimensional space, where the projected representations are highly linearly correlated [15]. As shown in Fig. 1, $\boldsymbol{h}_c$ and $\boldsymbol{\delta}$ can be seen as the data of two views from a $2MN$-dimensional space and an N-dimensional space, respectively. The DCCA structure projects them into an S-dimensional space $\mathcal{S}$ by $f_m(\cdot)$ and $g_e(\cdot)$, respectively, such that the projected data $f_m(\boldsymbol{h}_c)$ and $g_e(\boldsymbol{\delta})$ are highly linearly correlated. Such a function block embeds the information of selected antennas into a compressed channel vector, and thus, can model the underlying relevance between $\boldsymbol{H}$ and $\boldsymbol{\delta}$.

Autoencoder can recover the original data from a highly compressed form in a low-dimensional space [19]. As seen in Fig. 1, the autoencoder shares the encoding network $g_e(\cdot)$ with DCCA to compress the optimal AS vector $\boldsymbol{\delta}$ into a low-dimensional representation $g_e(\boldsymbol{\delta})$ in $\mathcal{S}$. The decoding network $g_d(\cdot)$ then predicts the AS vector directly from the low-dimensional representation. Since $f_m(\boldsymbol{h}_c)$ and $g_e(\boldsymbol{\delta})$ are highly correlated in $\mathcal{S}$ by the DCCA block, the decoding network $g_d(\cdot)$ can predict the AS vector from the output of $f_m(\cdot)$ for online instances. In sum, the MLL network in Fig. 1 can extract the relationship between the channel instance and the corresponding AS vector in a low-dimensional space by DCCA, and then recover the AS vector from the compressed data by the autoencoder.

C. Loss Function

The entire MLL model is treated as an end-to-end deep neural network which is trained in the supervised style. To carry out supervised training, a dataset consisting of a large number of labelled samples needs to be collected beforehand. For the considered problem, the dataset is denoted as $D = \{(\boldsymbol{h}_c(j), \boldsymbol{\delta}(j))\}_{j=1}^{R}$, where tuple $(\boldsymbol{h}_c(j), \boldsymbol{\delta}(j))$ is the j-th sample consisting of the vector-form channel instance $\boldsymbol{h}_c(j)$ and the AS vector $\boldsymbol{\delta}(j)$. We further denote the concatenated form of the dataset as $\boldsymbol{H}_{con} = [\boldsymbol{h}_c(1), \boldsymbol{h}_c(2), \ldots, \boldsymbol{h}_c(R)] \in \mathbb{R}^{2MN \times R}$ and $\boldsymbol{\Delta}_{con} = [\boldsymbol{\delta}(1), \boldsymbol{\delta}(2), \ldots, \boldsymbol{\delta}(R)] \in \mathbb{R}^{N \times R}$.

Based on the labelled dataset, the MLL network can optimize the trainable parameters based on the loss function, which is defined as

$$\mathcal{L} = \mathcal{L}_1(f_m, g_e) + \alpha \mathcal{L}_2(g_e, g_d), \tag{10}$$

where $\mathcal{L}_1$ is the loss at $\mathcal{S}$ measuring the correlation between $f_m(\boldsymbol{h}_c)$ and $g_e(\boldsymbol{\delta})$, $\mathcal{L}_2$ is the loss at the output of the autoencoder which assesses the closeness between the predicted AS vector $\tilde{\boldsymbol{\delta}} = g_d(f_m(\boldsymbol{h}_{in}))$ and the input AS vector $\boldsymbol{\delta}$, and α is a hyperparameter that determines the weight of each part. The variables are the trainable parameters of the component networks, denoted by f_m, g_e and g_d, respectively. The details of the loss function are presented as follows.

The objective of DCCA can be transformed into a distance minimization problem where the distance between two matrices is measured by Frobenius norm [20],

$$\begin{aligned}\min_{f_m, g_e} \quad & \|f_m(\boldsymbol{H}_{con}) - g_e(\boldsymbol{\Delta}_{con})\|_F^2 \\ s.t. \quad & f_m(\boldsymbol{H}_{con})\left[f_m(\boldsymbol{H}_{con})\right]^T = \boldsymbol{I}, \\ & g_e(\boldsymbol{\Delta}_{con})\left[g_e(\boldsymbol{\Delta}_{con})\right]^T = \boldsymbol{I}.\end{aligned} \tag{11}$$

Problem (11) can be rewritten with Lagrange multipliers as

$$\mathcal{L}_1 = \mathrm{tr}\left(\boldsymbol{C}_1^T\boldsymbol{C}_1\right) + \lambda\,\mathrm{tr}\left(\boldsymbol{C}_2^T\boldsymbol{C}_2 + \boldsymbol{C}_3^T\boldsymbol{C}_3\right), \tag{12}$$

where

$$\begin{aligned}\boldsymbol{C}_1 &= f_m(\boldsymbol{H}_{con}) - g_e(\boldsymbol{\Delta}_{con}), \\ \boldsymbol{C}_2 &= f_m(\boldsymbol{H}_{con})\left[f_m(\boldsymbol{H}_{con})\right]^T - \boldsymbol{I}, \\ \boldsymbol{C}_3 &= g_e(\boldsymbol{\Delta}_{con})\left[g_e(\boldsymbol{\Delta}_{con})\right]^T - \boldsymbol{I}.\end{aligned} \tag{13}$$

For the autoencoder, we utilize a label-correlation sensitive loss function to measure the difference between $\boldsymbol{\delta}$ and $\tilde{\boldsymbol{\delta}}$, as well as to exploit the interdependency among antennas, which is given by [21]

$$\mathcal{L}_2 = \frac{1}{K(N-K)} \sum_{j=1}^{R} \sum_{(p,q)\in\mathcal{P}\times\mathcal{Q}} \exp\left[-\left(\tilde{\boldsymbol{\delta}}_p(j) - \tilde{\boldsymbol{\delta}}_q(j)\right)\right], \tag{14}$$

where $\mathcal{P} = \{p\,|\,\boldsymbol{\delta}_p(j) = 1\}$ and $\mathcal{Q} = \{q\,|\,\boldsymbol{\delta}_q(j) = 0\}$ respectively denote the sets of selected and unselected antennas for the j-th sample. The term $\left(\tilde{\boldsymbol{\delta}}_p(j) - \tilde{\boldsymbol{\delta}}_q(j)\right)$ is a measurement of the ”distance” between the output of the neural network for any pair (p, q) consisting of a selected antenna p and an unselected antenna q. Minimizing $\mathcal{L}_2$ is equal to maximizing the overall ”distance” between the selected group and the unselected group. Therefore, the difference between the input and predicted AS vector is minimized, while at the same time, the interdependency among antennas is exploited owing to the pairwise evaluation metric.

Note that the model can also be trained by using traditional loss functions like standard mean squared error and cross-entropy loss. However, these loss functions fail to model the interdependency among labels as they only concentrate on independent label discrimination [12]. Thus, they are not desirable for the considered AS problem which has combinatorial complexity at the output space.

D. Implementation Details

1) Dataset Generation: Since an exhaustive search for the optimal AS vector is computationally prohibitive in massive MIMO systems, the aforementioned near-optimal convex relaxation based approach [4] is utilized to generate training samples. For each considered scenario with different K and ρ, we separately generate 10,000 labeled samples, and then randomly split the dataset into three different parts, i.e., 60%, 20%, 20%, for the purpose of training, validation and testing, respectively.

2) Hyperparameter Selection: The number of neurons in each layer of $f_m(\cdot)$ and $g_e(\cdot)$ is respectively given by 512, 256 and S, and the number of neurons in each layer of $g_d(\cdot)$ is respectively given by 256, 512 and S. We define the dimensional reduction ratio of the autoencoder as $\gamma = S/N$. By trial and error, the hyperparameter γ is set as 0.8, the hyperparameter α in Eq. (10) is set as 2, and hyperparameter λ in Eq. (12) is given by 0.5.

3) Network Training: The trainable parameters are updated by using the stochastic gradient descent method with momentum, where the initial learning rate is 1×10^{-4} and the size of each mini batch is 500. The learning rate is decayed by 2% each epoch, and the value of momentum is given by 0.99.

E. Complexity Analysis

In the study of deep learning, the computational complexity of neural networks is usually measured in terms of the floating-point operations (FLOPs). The proposed MLL network is mainly composed of FC layers, while the CNN in [14] mainly comprises both convolution and FC layers. For each FC layer, the number of FLOPs is given by $(2N_I - 1)N_O$, where N_I and N_O denote the input and output dimensions, respectively [22]. For each convolution layer, the number of FLOPs is given by $2HW(C_{in}K_w^2 + 1)C_{out}$, where H, W, and C_{in} respectively denote the height, width and number of channels of the input feature map, K_w is the width of the convolutional kernel, and C_{out} is the number of output channels [22]. Specifically, the number of FLOPs of the proposed MLL network is 1.16 million, while the number of FLOPs of the CNN in [14] is 34.6 million. Therefore, our proposed MLL network has a lower complexity as compared to the CNN.

For the convex relaxation based methods [4], the computational complexity is highly determined by the efficiency of solving the relaxed problem, for which we adopt two different approaches for comparison, i.e., the general-purpose CVX solver [23] and the low-complexity Frank-Wolfe algorithm proposed in [7]. The CVX solver incurs a complexity of at least $O(N^3KM)$ [3], while the Frank-Wolfe algorithm achieves a linear complexity in N, expressed as $O\left(\frac{N}{\epsilon}\left(M^2\left(M+1\right)+K\right)\right)$, where ϵ is the relative tolerance factor determining when the iteration terminates [7]. To show the complexity of deep learning based and convex relaxation based methods straightforwardly, in the following section, we will give the elapsed time of these methods.

IV. Simulation Results

We validate our proposed method by using the *QuaDRiGa* channel model [24]. Specifically, the base station employs an 8×4 panel element array, where each element consists of two orthogonal antennas. Hence, the total number of available antennas is given by $N = 64$. The number of users is given by $M = 4$, and they are randomly distributed in a circle area with a radius of 100 meter. Each user is equipped an omnidirectional antenna. The propagation environment is set as urban micro cell with line-of-sight conditions. The center frequency is set as 2.53 GHz. All simulations are implemented in MATLAB

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3087132, IEEE Transactions on Vehicular Technology

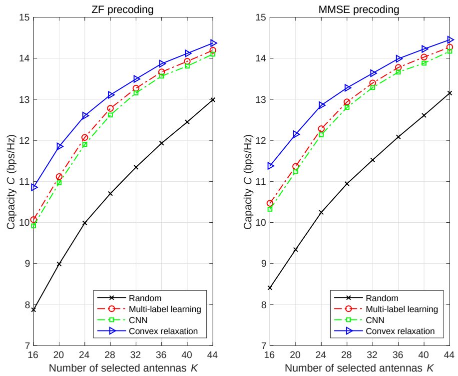

Fig. 2. Downlink capacity as a function of the number of selected antennas K when ρ = 5dB.

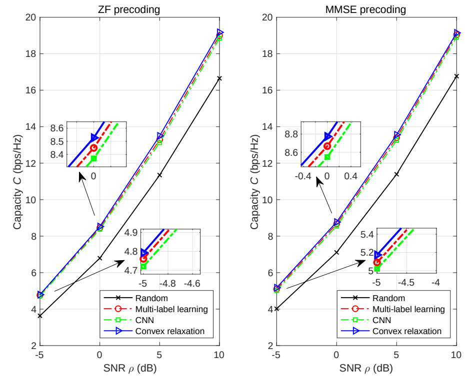

Fig. 3. Downlink capacity as a function of interference-free per-user SNR ρ when K = 32.

and conducted with 16G RAM on the same Intel Core i7-9750H CPU.

Our proposed MLL-based method is compared with three benchmark algorithms: convex relaxation, CNN and random selection. The convex relaxation method is originally proposed in [4] and has shown a near-optimal capacity performance in field trials. Nevertheless, it comes at the cost of a relatively high complexity due to the large-scale antenna array. We adopt two different methods for the relaxed convex programming problem, i.e., the general-purpose CVX solver [23] and the low-complexity Frank-Wolfe algorithm proposed in [7]. The CNN-based AS method [14] is trained and tested on the same dataset and simulation environment as our proposed MLL network for fair comparison. The random selection algorithm has reasonable performance with almost constant complexity. Note that exhaustive search is not simulated as it is computational prohibitive for massive MIMO.

Fig. 2 shows the downlink capacity as a function of the number of selected antennas K when ρ = 5dB, under both ZF and MMSE precoding schemes. The capacity increases with K for all considered methods as the spatial degrees of freedom increase, but the relative gaps between the algorithms decrease with K. When all antennas are selected, i.e., $K = N$, all four methods converge to the classic massive MIMO system without AS, and achieve the same capacity. As we see, the proposed MLL-based method significantly outperforms the random selection method and achieves $94\% - 98.8\%$ capacity of the convex relaxation based method as K increases from 16 to 44. The results indicate that the MLL-based method can effectively learn the relationship between the channel instance $\boldsymbol{H}$ and the AS vector $\boldsymbol{\delta}$, and achieve a comparable performance relative to the prevailing convex relaxation based method in networks with various number of selected antennas. We see that the MLL-based method also demonstrates better performance than the CNN-based method in all cases since the DCCA and autoencoder structure can effectively capture both the feature of channel instances and the interdependency among selected antennas.

In Fig. 3, we present the downlink capacity as a function of the interference-free per-user SNR ρ when K = 32, under both ZF and MMSE precoding schemes. As we see, the performance of the MLL-based method significantly outperforms the random selection method, and the performance gap compared to the convex relaxation based method is less than 2% for the simulated SNR values varying from -5dB to 10dB. The results indicate that the MLL-based method can effectively perform antenna selection in massive MIMO systems with different values of SNR, and achieve a comparable performance relative to the convex relaxation based method. The MLL-based method also outperforms the CNN-based method in both low and high SNR regions.

TABLE I. Computation time comparison

No. of selected antennas K	Average computation time (ms)			
	CVX	Frank-Wolfe	CNN-based	MLL-based
32	1190.58	513.7	3.091	0.1602
36	1201.83	517.5	3.103	0.1613
40	1224.90	523.6	3.116	0.1633
44	1273.51	534.7	3.111	0.1690

In Table I, we compare the online computation time of the considered methods. The convex relaxation method is implemented via both the standard CVX solver and the low-complexity Frank-Wolfe algorithm. Both deep learning based methods, i.e., CNN and the proposed MLL network, need to be trained offline before online deployment. The training process consists of two parts: generating the labeled dataset and training the neural network with labeled samples. As for the first part, training an MLL network requires to generate a 10,000-sample dataset, which takes about 18 minutes for the considered 64×4 multi-user massive MIMO system on the simulation platform when parallel computing is enabled. As for the second part, the training process normally takes about 1-2 minutes to converge. Once trained, the neural network can be deployed online for fast inference given unseen input channel instances. As shown in the table, the online computation time of the trained MLL network is three orders of magnitude lower than both the CVX and Frank-Wolfe realizations of the convex relaxation based method, and is about 5% of that of the CNN-based method in the considered scenarios. Note that the online operation of the trained MLL network can be further accelerated by using GPUs. Since antenna selection is a real-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3087132, IEEE Transactions on Vehicular Technology

TABLE II. Generalization ability to topology mismatch

Training topology	Testing topology	Performance	Relative loss
M = 4	M = 1	96.7%	-1.8%
	M = 2	97.1%	-1.3%
	M = 3	97.8%	-0.5%
	M = 4	98.2%	0%

time task whose computation time should generally not exceed the channel coherence time, the significant speedup provided by the proposed MLL network makes it a promising candidate for practical implementation.

In Table II, we show the generalization ability of the MLL network to topology mismatch when ρ = 5dB and K = 32. The performance and the relative loss compared to specialized neural networks are normalized by the capacity achieved by the convex relaxation method. We first train the MLL network with data collected under M = 4 users, and then test the performance when there exists mismatch in terms of the user number, which can be realized by padding an appropriate number of zeros to the input. As shown in the table, only slight performance degradation is observed, which demonstrates the generalization ability of the neural network to mild topology mismatch. It can also be observed that the relative performance loss slightly increases when the topology mismatch becomes more apparent. These results indicate that each cell may only need to train a single MLL network to handle the changeable number of users, whose input dimension should be set to be consistent with the maximum number of supportable users. When the number of supportable users is fairly large, the base station can train and store MLL networks to respectively deal with the case of low, medium and high traffic, so as to compensate for the performance loss. We also note that there have been some recent advances in designing universal learning models that are quite robust to any topology settings [25, 26]. As for future work, it is of interest to investigate how they can be combined with the proposed MLL network to further enhance the performance.

V. Conclusion

In this paper, we have proposed a deep neural network based AS scheme for multi-user massive MIMO systems. The proposed method is based on the framework of multi-label learning, in which deep canonical correlation analysis and an autoencoder are combined in a unified network structure to extract and leverage the low-dimensional features of channel matrix, as well as the underlying interdependency among the selected antennas. Simulation results show that, compared with the convex relaxation based method, the proposed MLL-based can achieve comparable capacity in typical network settings, while the computation time is significantly reduced by orders of magnitude.

Acknowledgement

The authors would like to thank the editors and the anonymous reviewers, whose invaluable comments helped improve the presentation of this paper substantially.

References

[1] E. G. Larsson *et al.*, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.
[2] A. F. Molisch and M. Z. Win, "MIMO systems with antenna selection," *IEEE Microw. Mag.*, vol. 5, no. 1, pp. 46–56, Mar. 2004.
[3] Z. Kuai and S. Wang, "Thompson sampling-based antenna selection with partial CSI for TDD massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7533–7546, Dec. 2020.
[4] X. Gao *et al.*, "Massive MIMO in real propagation environments: Do all antennas contribute equally?" *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 3917–3928, Nov. 2015.
[5] B. Makki *et al.*, "A genetic algorithm-based antenna selection approach for large-but-finite MIMO networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6591–6595, Jul. 2017.
[6] J. C. Marinello Filho *et al.*, "Antenna selection for improving energy efficiency in XL-MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 305–13 318, Nov. 2020.
[7] A. Konar and N. D. Sidiropoulos, "A simple and effective approach for transmit antenna selection in multiuser massive MIMO leveraging submodularity," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4869–4883, Sept. 2018.
[8] M. O. K. Mendona *et al.*, "Antenna selection in massive MIMO based on greedy algorithms," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1868–1881, Mar. 2020.
[9] Z. Abdullah *et al.*, "Efficient low-complexity antenna selection algorithms in multi-user massive MIMO systems with matched filter precoding," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2993–3007, Mar. 2020.
[10] J. Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2241–2244, Nov. 2016.
[11] P. Yang *et al.*, "Antenna selection for MIMO system based on pattern recognition," *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 34–39, Feb. 2019.
[12] M. S. Ibrahim *et al.*, "Learning-based antenna selection for multicasting," in *Proc. IEEE SPAWC*, Kalamata, Greece, Jun. 2018.
[13] T. X. Vu *et al.*, "Machine learning based antenna selection and power allocation in multi-user MISO systems," in *Proc. WiOPT*, Avignon, France, Jun. 2019.
[14] A. M. Elbir and K. V. Mishra, "Joint antenna selection and hybrid beamformer design using unquantized and quantized deep learning networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1677–1688, Mar. 2020.
[15] G. Andrew *et al.*, "Deep canonical correlation analysis," in *Proc. ICML*, Atlanta, GA, USA, Jun. 2013.
[16] E. Bjrnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure," *IEEE Signal Process. Mag.*, vol. 31, no. 4, pp. 142–148, Jul. 2014.
[17] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.
[18] C. Nwankpa *et al.*, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, Nov. 2018.
[19] C.-K. Yeh *et al.*, "Learning deep latent spaces for multi-label classification," in *Proc. AAAI*, San Francisco, CA, USA, Feb. 2017.
[20] D. R. Hardoon, S. Szedmk, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.
[21] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.
[22] P. Molchanov *et al.*, "Pruning convolutional neural networks for resource efficient inference," in *Proc. ICLR*, Toulon, France, Apr. 2017.
[23] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.
[24] S. Jaeckel *et al.*, "Quadriga: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.
[25] H. Lee, M. Gitnyk, and J. Jeong, "Deep MIMO autoprecoder," in *Proc. IEEE ICC*, Dublin, Ireland, Jun. 2020.
[26] H. Lee, M. Girnyk, and J. Jeong, "Deep reinforcement learning approach to MIMO precoding problem: Optimality and robustness," *arXiv preprint arXiv:2006.16646*, Jun. 2020.