# Transfer Learning for Semi-Supervised Automatic Modulation Classification in ZF-MIMO Systems

Yu Wang, *Student Member*, *IEEE*, Guan Gui, *Senior Member*, *IEEE*, Haris Gacanin, *Senior Member*, Tomoaki Ohtsuki, *Senior Member*, *IEEE*, Hikmet Sari, *Fellow*, *IEEE*, and Fumiyuki Adachi, *Life Fellow*, *IEEE*

***Abstract*—Automatic modulation classification (AMC) is an essential technology for the non-cooperative communication systems, and it is widely applied into various communications scenarios. In the recent years, deep learning (DL) has been introduced into AMC due to its outstanding identification performance. However, it is almost impossible to implement previously proposed DL-based AMC algorithms without large number of labeled samples, while there are generally few labeled sample and large unlabel samples in the realistic communication scenarios. In this paper, we propose a transfer learning (TL)-based semi-supervised AMC (TL-AMC) in a zero-forcing aided multiple-input and multiple-output (ZF-MIMO) system. TL-AMC has a novel deep reconstruction and classification network (DRCN) structure that consists of convolutional auto-encoder (CAE) and convolutional neural network (CNN). Unlabeled samples flow from CAE for modulation signal reconstruction, while labeled samples are fed into CNN for AMC. Knowledge is transferred from the encoder layer of CAE to the feature layer of CNN by sharing their weights, in order to avoid the ineffective feature extraction of CNN under the limited labeled samples. Simulation results demonstrated the effectiveness of TL-AMC. In detail, TL-AMC performs better than CNN-based AMC under the limited samples. What's more, when compared with CNN-based AMC trained on massive labeled samples, TL-AMC also achieved the similar classification accuracy at the relative high SNR regime.**

***Index Terms*—Automatic modulation classification, deep learning, convolutional neural network, transfer learning, multiple-input and multiple-output.**

## I. Introduction

Automatic modulation classification (AMC) plays an essential role for the demodulation of signals in many non-cooperative communications systems [1] and beyond fifth generation (B5G) wireless communications systems [2], [3]. AMC originates from the military scenario, which is widely applied into electronic warfare for the analysis of the intercepted signals from enemy [4]. Then, AMC has been gradually introduced into civilian applications, such as spectrum resource monitoring [1], [5], link adaptation (LA) [4], [6], and cognitive radio (CR) [7]. In general, AMC is modeled as a typical pattern recognition problem, and a series of traditional methods, based on manmade features and machine learning classifier [8], have been proposed and realized. However, their recognition performance still require to be improved.

Recently, deep learning (DL), as an effective and powerful tool of classification and regression tasks [9]–[14], has been applied for solving various communications problems [15]–[17], such as beam management [18], system optimization [19]–[24], resource allocation [25]–[29], channel codec [30] and so on. For the same reasons, DL is introduced into AMC, and many DL-based AMC algorithms have been proposed [31]–[36]. However, these algorithms are based on a large number of labeled samples, and they may be not suitable for the real communication scenarios, where labeled samples are few. The learning algorithms, which are based on a large number of unlabeled samples and few labeled samples, are denoted as semi-supervise algorithm.

Hence, inspired by that transfer learning (TL) [37] is generally applied for semi-supervised learning, we propose a TL-based semi-supervised AMC (TL-AMC) method in ZF equalization-aided multiple-input and multiple-output (MIMO) systems [35]. The proposed TL-AMC method can make full use of both labeled samples and unlabeled samples. Specifically, we introduce a TL structure with two pipeline, and the one is a CNN for classification of the few labeled samples, while the other is a convolutional auto-encoder (CAE) to reconstruct the massive unlabeled samples. The encoder layer in CAE shares the knowledge (i. e., weights)with the convolutional layer in CNN to help CNN overcome the problems of ineffective training and overfitting, which are caused by few labeled samples. The simulation results demonstrated the effectiveness of the proposed TL-AMC algorithm.

The rest of this paper is organized as follows. In Section II, we overview the related work about AMC algorithms. In Section III, we introduce system model, signal model and the process of dataset generation. Then, in IV, we introduce the proposed semi-supervised TL-AMC algorithm, and the CNN-based AMC algorithm, as a comparison, is briefly introduced.

This work was supported by the Project Funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant TC190A3WZ-2, National Natural Science Foundation of China under Grant 61901228 and 61671253, the Jiangsu Specially Appointed Professor under Grant RK002STP16001, the Innovation and Entrepreneurship of Jiangsu High-level Talent under Grant CZ0010617002, the Six Top Talents Program of Jiangsu under Grant XYDXX-010, the 1311 Talent Plan of Nanjing University of Posts and Telecommunications. *(Corresponding author: Guan Gui.)*

Y. Wang, G. Gui, H. Sari are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: {1018010407, guiguan, hikmet}@njupt.edu.cn)

H. Gacanin is with the Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 55-52062, Germany (e-mail: harisg@ieee.org)

T. Ohtsuki is with the Department of Information and Computer Science, Keio University, Yokohama 223-8521, Japan (e-mail: ohtsuki@ics.keio.ac.jp)

F. Adachi is with the Research Organization of Electrical Communication (ROEC), Tohoku University, Sendai 980-8577 Japan (e-mail: adachi@ecei.tohoku.ac.jp)

TABLE I
THE MAIN SYMBOLS IN THIS PAPER.

| Symbol | Explanation |
|---|---|
| $N_r$ | The number of receiving antennas |
| $N_t$ | The number of transmitting antennas |
| $\mathbf{H}$ | Complex MIMO channel matrix |
| $\boldsymbol{y}_n$ | Received signal vector |
| $\boldsymbol{x}_n$ | Transmitted modulation signal vector |
| $\boldsymbol{w}_n$ | Additive white Gaussian noise (AWGN) |
| $\hat{\boldsymbol{y}}_n$ | Received signal vector processed by ZF equalizer |
| $ZF(\mathbf{H})$ | ZF equalization matrix |
| $\mathcal{D}$ | Generated random symbol vector |
| $K$ | Length of generated random symbol vector |
| $Mod(\cdot)$ | Modulation operation |
| $\mathbf{X}$ | Transmitted signal matrix |
| $\mathbf{Y}$ | Received signal matrix |
| $\mathbf{W}$ | Noise matrix |
| $Vec(\cdot)$ | Vectorization operation |
| $Re(\cdot), Im(\cdot)$ | Real part, Imaginary part |
| $\{P(m\|IQ)\}_{m\in\mathcal{M}}$ | Probability distribution function (PDF) |
| $\{s_i, l_i\}_{i=1}^{N_C}$ | Labeled dataset with $N_C$ samples |
| $\{s_j\}_{j=1}^{N_R}$ | Unlabeled dataset with $N_R$ samples |
| $\mathcal{D}_S/\mathcal{X}_S/\mathcal{T}_S/\mathcal{F}_S$ | Source dimension/dataset/task/mapping function |
| $\mathcal{D}_T/\mathcal{X}_T/\mathcal{T}_T/\mathcal{F}_T$ | Target dimension/dataset/task/mapping function |
| $f_{CNN}/f_{CAE}$ | Mapping function of CNN/CAE |
| $g_F/g_C$ | Mapping function of feature/classification layer |
| $g_E/g_D$ | Mapping function of encoder/decoder layer |
| $\mathbf{\Theta}_{CNN}/\mathbf{\Theta}_{CAE}$ | Parameters of CNN/CAE |
| $\mathbf{\Theta}_F/\mathbf{\Theta}_C$ | Parameters of feature/classification layer |
| $\mathbf{\Theta}_E/\mathbf{\Theta}_D$ | Parameters of encoder/decoder layer |
| $\mathcal{L}_{CCE}/\mathcal{L}_{MSE}$ | Loss function for the CNN/CAE |
| $\lambda_{CAE}$ | Factor of MSE loss function for CAE |
| $\gamma$ | Learning rate |
| $B$ | Batch size |

Next, various simulation results are provided to compare their performance in the section V. Finally, the conclusion of this paper is given in Section VI. In addition, the main symbols are listed in Tab. I.

## II. RELATED WORKS

In this section, we will introduce the previously proposed supervised and semi-supervised AMC algorithms based on DL in both single input and single output (SISO) systems and MIMO systems, respectively.

### A. Supervised AMC Methods

DL-based AMC was firstly proposed for SISO system. Specifically, T. J. O'Shea, *et al.* [31] firstly proposed a novel convolutional neural network (CNN)-based AMC algorithm under the Rician fading environments, whose training samples are combined with the in-phase and quadrature (IQ) component of signals, i.e., IQ sample. Its excellent performance is far beyond the performance of traditional AMC algorithms, and other DL-based AMC algorithms stemed from it. Based on [31], Z. Zhang, *et al.* [32] proposed a feature fusion algorithm, and authors attempted to fuse various images and handcrafted features of signals into more robust and effective features for better classification performance. In addition, different form the former two papers, S. Hu, *et al.* [33] adopted a long short-term memory network with attention mechanism for AMC, and the AMC algorithm is effectively implemented in the condition of additive white Gaussian noise (AWGN) and two non-Gaussian noises.

Regarding to various AMC algorithms in the SISO system, AMC in the MIMO systems is also concerned. Y. Wang, *et al.* [34] proposed a CNN-based cooperative AMC (Co-AMC), which give the final modulation type decision via the sub-results from each receiving antenna and various cooperative decision rules. What's more, based on paper [34], G. Gui, *et al.* [35] adopted zero forcing (ZF) equalization with channel statement information (CSI) to improve the performance of AMC. In detail, ZF equalizer is applied to pre-process the received signals, and then the corresponding modulation types are identified by pre-processed signals and CNN. Different from the simple MIMO system with a uncorrelated Rayleigh fading channel in above two papers, M. H. Shah and X. Dang [36] considered a more complex system: space-time-block-codes (STBC)-MIMO system. In this paper, they proposed two neural network structures: sparse auto-encoder (SAE)-based deep neural network (DNN) and radial basis function network (RBFN) for AMC, and they also studied the impact of the CSI estimation errors on the performance of AMC.

### B. Semi-supervised AMC Methods

DL models in the above AMC algorithms are trained on a large number of labeled samples, and these algorithms can be considers as the supervised algorithm. However, there are generally a small number of labeled samples with massive unlabeled samples in the actual communication scenarios. Under this condition, the previous proposed DL models can not perform well and effectively. Thus, few shot learning [38] or semi-supervised algorithms are more suitable for AMC in this case.

However, few researches are involved into the semi-supervised AMC in SISO system [39]–[41], and there are almost no papers to report semi-supervised AMC methods in MIMO systems. T. J. O'Shea, *et al.* [39] firstly proposed a DL-based semi-supervised AMC algorithm, which consists of an auto-encoder (AE) for nonlinear feature extraction and various clustering algorithms with few labeled samples for classification, but it has an unsatisfactory performance. Y. Tu, *et al.* [40] and M. Li *et al.* [41] proposed two auxiliary classifier generative adversarial network (ACGAN)-based semi-supervised AMC algorithms, respectively. Their methods are to create more samples by ACGAN for the effective training of supervised AMC, and the main difference between them is the sample form, and the former is the format of constellation diagram, while the sample form for the latter is the same as [31].

## III. SYSTEM MODEL, SIGNAL MODEL AND DATASET GENERATION

### A. System Model

In this paper, when transmitting the modulation signals by transmitter, the receiver in the non-cooperative communication system must first recognize the modulation type of this received signal by the AMC algorithm, and then it can demodulate this signal, because there are no protocol between

transmitter and receiver in the non-cooperative communication system. In addition, a typical ZF-MIMO system is considered with $N_t$ antennas in the transmitter and $N_r$ antennas in the receiver ($N_r \geq N_t$), and the receiver is equipped with ZF equalizer for improving the performance of the system [42]. The whole system structure is shown in Fig. 1.

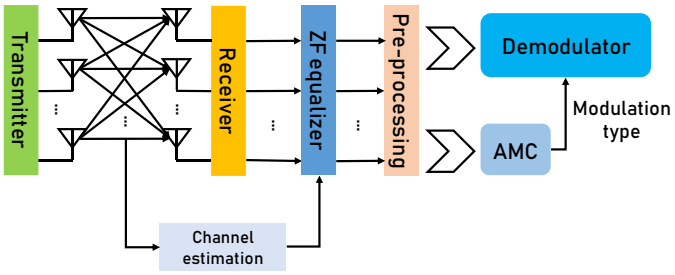


Fig. 1. The system structure of AMC-aided ZF-MIMO system. Expect the transmitter and the receiver equipped with ZF equalizer, AMC module is applied to identify the modulation type for the next demodulation.

### B. Signal Model

Here, we consider a uncorrelated Rayleigh flat-fading and time invariant MIMO channel [42]. The complex MIMO channel matrix is denoted as $\mathbf{H}$, whose dimensionality is $N_r \times N_t$, and each element of the channel matrix is i. i. d, and obeys zero-mean and unit variance complex Gaussian distribution. Thus, the received MIMO signal at time $n$ can be model as

$$\boldsymbol{y}_n = \mathbf{H}\boldsymbol{x}_n + \boldsymbol{w}_n, \tag{1}$$

where $\boldsymbol{y}_n = [y_n^1, y_n^2, ..., y_n^{N_r}]^T$ is the $(N_r \times 1)$ received signal vector, which is perfectly sampled without carrier frequency offset and phase offset, and $y_n^i, i \in [1, N_r]$ is the received symbol at the $i$-th antenna; $\boldsymbol{x}_n = [x_n^1, x_n^2, ..., x_n^{N_t}]^T$ is the $(N_t \times 1)$ transmitted signal vector with a certain modulation type, and , and $x_n^j, j \in [1, N_t]$ is the transmitted symbol at the $j$-th antenna; $\boldsymbol{w}_n$ is the AWGN with size $(N_r \times 1)$, and $\boldsymbol{w}_n \sim \mathcal{CN}(0, N_0\mathbf{I}_{N_r})$.

Due to the application of ZF equalizer, the received signal can be written as

$$\hat{\boldsymbol{y}}_n = ZF(\mathbf{H})\boldsymbol{y}_n = \boldsymbol{x}_n + ZF(\mathbf{H})\boldsymbol{w}_n, \tag{2}$$

where $ZF(\mathbf{H}) = \mathbf{H}^\dagger = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$, and $(\cdot)^\dagger$ and $(\cdot)^H$ are respectively the pseudo inverse operation and complex conjugate transpose [43].

### C. Dataset Generation

Based on the above signal model, we can generate dataset for training and test, and the specified steps are shown in Fig. 2, and the dimensionality of the generated data after each conversion or operation are also listed in Fig. 2. The former two steps are to generate random symbol vector $\mathcal{D}$ with size $1 \times K$ ($K = 128$), and modulate this data to achieve $Mod(\mathcal{D})$. In this paper, we consider a popular and typical modulation type pool: $\mathcal{M}$ = {BPSK, QPSK, 8PSK, 16QAM} in both SISO systems and MIMO systems [1], [33]–[35]. Then, the energy of $Mod(\mathcal{D})$ is normalized to $\overline{Mod(\mathcal{D})}$, i. e., $\sum_{k=1}^{K} |\overline{Mod(\mathcal{D})}|^2 = 1$. This operation is to fairly recognize different modulation type. Next, $\overline{Mod(\mathcal{D})}$ is reshaped into a matrix $\mathbf{X}$ with size $N_t \times \frac{K}{N_t}$, and the $n$-th column of $\mathbf{X}$ is equal to $\boldsymbol{x}_n$, $1 \leq n \leq \frac{K}{N_t}$, i. e., $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{\frac{K}{N_t}}]$.

When passing through the MIMO channel and adding AWGN, the transmitted matrix $\mathbf{X}$ can be converted into the $(N_r \times \frac{K}{N_t})$ received matrix $\mathbf{Y} = \mathbf{HX} + \mathbf{W}$, where the $n$-th column of $\mathbf{Y}$ and $\mathbf{W}$ are respectively $\boldsymbol{y}_n$ and $\boldsymbol{w}_n$, $1 \leq n \leq \frac{K}{N_t}$, i. e., $\mathbf{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_{\frac{K}{N_t}}]$ and $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{\frac{K}{N_t}}]$. Finally, after ZF equalization and vectorization, we separate the $1 \times K$ vector $S = Vec[ZF(\mathbf{H})\mathbf{Y}]$, where $Vec(\cdot)$ is the vectorization operation, into its real part $Re(S^T)$ and imaginary part $Im(S^T)$. Then, $Re(S^T)$ and $Im(S^T)$ are combined into a $K \times 2$ real-value matrix, which is denoted as $IQ = [Re(S^T), Im(S^T)]$, because the real and imaginary part are the in-phase (I) and quadrature (Q) component, respectively [35].

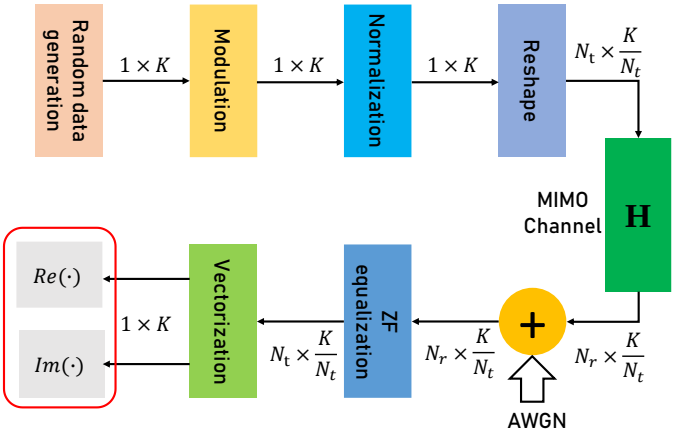


Fig. 2. The detailed steps of dataset generation and the changes of dimensionality after each step. It contains nine steps form the initial random data generation to IQ sample.

## IV. The Proposed Semi-supervised TL-AMC Algorithm

Recently, there have been many researches focusing on the supervised DL-based AMC algorithms, and most of researches model AMC as a maximum a posteriori (MAP)-based classification problem [33], which can be expressed as

$$m^* = \arg\max_{m \in \mathcal{M}} P(m|IQ), \tag{3}$$

where $m^*$ and $m$ is the predicted modulation type and the real modulation type, respectively. $\{P(m|IQ)\}_{m \in \mathcal{M}}$ is the probability distribution function (PDF) of the input $IQ$, and it is also the output of the DL models. What's more, it is well-known that huge high quality samples and accurate labels are required for the training of the supervised DL-based AMC algorithms. However, the corresponding labels of these samples are not usually achievable, and there are more commonly small amounts of correctly labeled samples with large amounts of unlabeled samples. Under this condition, the common supervised algorithms are almost ineffective, but

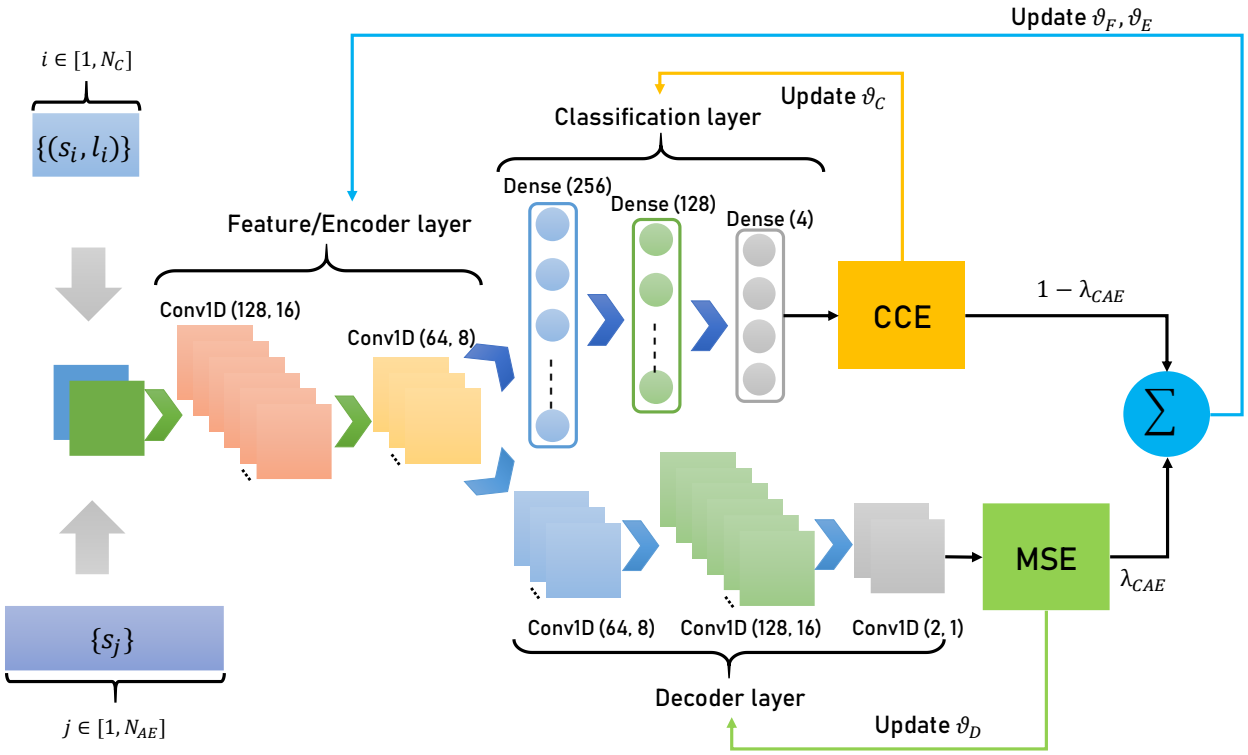


Fig. 3. TL-based semi-supervised AMC structure: DRCN. DRCN has two modules: CAE and CNN, where CAE is applied for the reconstruction of unlabeled sample $\{s_j\}_{j=1}^{N_R}$, and CNN is trained on the labeled samples $\{s_i, l_i\}_{i=1}^{N_C}$. What's more, the encoder layer of CAE and the feature layer of CNN have the same structure and share their weights with each other. In addition, "Conv1D" is a general convolutional layer and "Dense" is a common fully-connected layers.

semi-supervised DL algorithms may work. Here, we focus on the semi-supervised AMC algorithm in the case of the limited labeled samples.

Inspired by paper [44], we introduce a novel TL framework for the semi-supervised AMC algorithm in the ZF-MIMO system, which is denoted as TL-AMC. In the next section, we will introduce the TL-AMC algorithm from four aspects: problem description and assumed scenario, TL framework, loss function, and learning algorithm.

### A. Problem Description and Assumed Scenario

In recent years, DL-based AMC algorithms have achieved outstanding performances, but they are based on massive labeled modulation signals. However, the real communication situation is that there are few accurate labeled samples, but the number of unlabeled samples is huge. It is scarcely possible to directly train the DL-based AMC algorithms with the powerful performance, based on the few labeled samples. Hence, we adopt TL into AMC for improving the weak performance under the condition of few labeled samples.

What's more, for the simulation of this condition, we generate a dataset with a great quantity of samples, and it is assumed that only the small percent of samples are correctly labeled, while the remaining samples are unlabeled samples. The correctly labeled samples and their corresponding labels is denoted as $\{s_i, l_i\}_{i=1}^{N_C}$, and the unlabeled sample is referred as $\{s_j\}_{j=1}^{N_R}$, where $N_C$ and $N_R$ are the number of labeled and unlabeled samples, respectively, and $N_C \ll N_R$. Based on the assumed dataset, it is impossible to directly apply the labeled dataset $\{s_i, l_i\}_{i=1}^{N_C}$ to train a CNN-based AMC algorithm with high performance, because CNN can not extract high robust and effective features from the few available samples.

### B. TL Framework

*1) A Brief Introduction for TL:* We define source dimension, source data, source task, and source mapping function as $\mathcal{D}_S$, $\mathcal{X}_S$, $\mathcal{T}_S$, and $\mathcal{F}_S$, while $\mathcal{D}_T$, $\mathcal{X}_T$, $\mathcal{T}_T$ and $\mathcal{F}_T$, are target dimension, target data, and target task and target mapping function, respectively. According to paper [45], TL is defined as the algorithm to improve the learning of $\mathcal{F}_T(\cdot)$ in $\mathcal{D}_T = \{\mathcal{X}_T, \mathcal{F}_T(\mathcal{X}_T)\}$ by the application of the knowledge in $\mathcal{D}_S = \{\mathcal{X}_S, \mathcal{F}_S(\mathcal{X}_S)\}$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ [45].

*2) Deep Reconstruction and Classification Network (DRCN):* Here, $\{s_j\}_{j=1}^{N_R}$ is considered as the source data, and $\{s_i, l_i\}_{i=1}^{N_C}$ is denoted as the target dataset, while these two dataset are applied for modulation signal reconstruction and AMC, respectively. Thus, the TL framework is defined as deep reconstruction and classification network (DRCN), and the reconstruction task is to improve the performance of AMC under the condition of few labeled samples. The specified structure is shown in Fig. 3, which consists of two modules for two tasks, respectively. The above module can be considered as a typical CNN for AMC, while the below module, the convolutional auto-encoder (CAE), is generally applied for the reconstruction task.

The CNN module is applied for supervised AMC to establish the mapping function from modulation signal to modulation type, i. e., $f_{CNN}:\{s_i\}_{i=1}^{N_C}\rightarrow\{l_i\}_{i=1}^{N_C}$. The CNN module contains the feature layer (two convolution layer) $g_F$, which is to map modulation signal to feature space, i. e., $g_F:\{s_i\}_{i=1}^{N_C}\rightarrow F$. In addition, the classification layer (three fully-connected layers) $g_C$ is to map the feature space to the modulation type, i. e., $g_C:\mathcal{F}\rightarrow\{l_i\}_{i=1}^{N_C}$.

What's more, the encoder layer $g_E$ and decoder layer $g_D$ make up of the the CAE module $f_{CAE}:\{s_j\}_{i=1}^{N_R}\rightarrow\{s_j\}_{i=1}^{N_R}$, which is an unsupervised learning. The encoder layer has the same structure with the feature layer and they share weights of trainable parameters with each other, which means that their has the same feature space, and the mapping function can be represented as $g_E:\{s_j\}_{i=1}^{N_R}\rightarrow\mathcal{F}$. The decoder layer is to reconstruct the modulation signal form the feature space, i. e., $g_D:\mathcal{F}\rightarrow\{s_j\}_{i=1}^{N_R}$, and it contains three-layer deconvolution layer. Thus, the $f_{CNN}$ and $f_{CAE}$ can be decomposed as

$$f_{CNN}(s_i)=g_F\circ g_C(s_i), \tag{4}$$

$$f_{CAE}(s_j)=g_E\circ g_D(s_j), \tag{5}$$

where "$\circ$" is the composite operation, and the functions are executed from left to right.

The other specified parameters, such as the number of neurons and the size of filters, are listed in Fig. 3. Batch normalization [46] and dropout follow behind each available layer, except the last fully-connected layer and the last deconvolution layer, in order to prevent overfitting and accelerate the training process. Rectified linear unit (ReLU) function is as the activation function, following behind each available, while softmax function and parametric ReLU (PReLU) are the activation functions of the last fully-connected layer and the last deconvolution layer. These activation function can be written as

$$f_{ReLU}(x_{in})=max(0,x_{in}), \tag{6}$$

$$f_{Softmax}(x_{in}^i)=\frac{x_{in}^i}{\sum_j x_{in}^j}, \tag{7}$$

$$f_{PReLU}=max(\alpha x_{in},x_{in}), \tag{8}$$

where $\alpha$ is a learnable parameter.

### C. Loss Function

Here, the categorical cross entropy (CCE) and the mean square error (MSE) are as the loss functions for the classification task and reconstruction task. Set $\mathbf{\Theta}_{CNN}=\{\mathbf{\Theta}_F,\mathbf{\Theta}_C\}$ and $\mathbf{\Theta}_{CAE}=\{\mathbf{\Theta}_E,\mathbf{\Theta}_D\}$. The function of CCE and MSE are given as

$$\begin{aligned}\mathcal{L}_{CCE}(\{s_i,l_i\}_{i=1}^{N_C};\mathbf{\Theta}_{CNN})=\\ -\frac{1}{N_C}\sum_{i=1}^{N_C}l_i\log[f_{CNN}(s_i;\mathbf{\Theta}_{CNN})],\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{L}_{MSE}(\{s_j\}_{j=1}^{N_R};\mathbf{\Theta}_{CAE})=\\ -\frac{1}{N_R}\sum_{j=1}^{N_R}||f_{CAE}(s_j;\mathbf{\Theta}_{CAE})-s_i||_2^2,\end{aligned} \tag{10}$$

where $f_{CNN}$ and $f_{CAE}$ are the mapping functions of the CNN for AMC and the CAE for the reconstruction task, respectively; $\mathbf{\Theta}_F$, $\mathbf{\Theta}_C$, $\mathbf{\Theta}_E$ and $\mathbf{\Theta}_D$ are the trainable parameters of the feature layer, the classification layer, the encoder layer, and the decoder layer, respectively. It is noted that $\mathbf{\Theta}_F$ is identically equal to $\mathbf{\Theta}_E$, i. e., $\mathbf{\Theta}_F\equiv\mathbf{\Theta}_E$.

Thus, we define the trainable parameters in the DRCN as $\mathbf{\Theta}=\{\mathbf{\Theta}_F,\mathbf{\Theta}_C,\mathbf{\Theta}_E,\mathbf{\Theta}_D\}$. Thus, the final loss function is based on these two loss functions, and it can be expressed as

$$\begin{aligned}\mathcal{L}_{DRCN}(\{s_i,l_i\}_{i=1}^{N_C},\{s_j\}_{j=1}^{N_R};\mathbf{\Theta})=\\ (1-\lambda_{CAE})\mathcal{L}_{CCE}(\{s_i,l_i\}_{i=1}^{N_C};\mathbf{\Theta}_{CNN})\\ +\lambda_{CAE}\mathcal{L}_{MSE}(\{s_j\}_{j=1}^{N_R};\mathbf{\Theta}_{CAE})\\ +\lambda J(f_{CNN},f_{CAE};\mathbf{\Theta}),\end{aligned} \tag{11}$$

where $\lambda_{CAE}$ is the MSE loss factor for the balance the classification task and the reconstruction task, and $0\leq\lambda_{CAE}\leq 1$. When the first two terms are considered as the data loss, the final term $J(\cdot)$ can be believed as the penalty term to constrain the model complexity and avoid overfitting, where $\lambda$ $(0\leq\lambda\leq 1)$ is for the balance between the data loss and the model complexity.

### D. Learning Algorithm

Here, we adopt stochastic gradient descent (SGD) as the learning algorithm to minimize the loss function of DRCN, which can be written as

$$\mathbf{\Theta}^*=\arg\min_{\mathbf{\Theta}}\mathcal{L}_{DRCN}(\{s_i,l_i\}_{i=1}^{N_C},\{s_j\}_{j=1}^{N_R};\mathbf{\Theta}). \tag{12}$$

The training of the DRCN is based on a small batch, where the labeled samples for the batch training can be denoted as $\{s_i,l_i\}_{i=1}^{B}\subset\{s_i,l_i\}_{i=1}^{N_C}$ $(B\ll N_C)$, and the unlabeled samples for the batch training can be written as $\{s_j\}_{j=1}^{B}\subset\{s_j\}_{j=1}^{N_R}$ $(B\ll N_R)$. Thus, the updating of parameters can be expressed as

$$\begin{aligned}\mathbf{\Theta}_{CNN}^{new}\leftarrow\mathbf{\Theta}_{CNN}\\ -(1-\lambda_{CAE})\gamma\nabla_{\mathbf{\Theta}_{CNN}}[\mathcal{L}_{CCE}(\{s_i,l_i\}_{i=1}^{B};\mathbf{\Theta}_{CNN})]\\ -\lambda\gamma\nabla_{\mathbf{\Theta}_{CNN}}[J(f_{CNN},f_{CAE};\mathbf{\Theta})],\end{aligned} \tag{13}$$

$$\begin{aligned}\mathbf{\Theta}_{CAE}^{new}\leftarrow\mathbf{\Theta}_{CAE}\\ -\lambda_{CAE}\gamma\nabla_{\mathbf{\Theta}_{CAE}}[\mathcal{L}_{MSE}(\{s_j\}_{j=1}^{B};\mathbf{\Theta}_{CAE})]\\ -\lambda\gamma\nabla_{\mathbf{\Theta}_{CAE}}[J(f_{CNN},f_{CAE};\mathbf{\Theta})],\end{aligned} \tag{14}$$

where $\gamma$ is the learning rate and $\nabla$ is the partial derivative operation.

The specified algorithm is shown in Algorithm 1. It is noted that "**parafor**" represents parallel operation, which means that the parameters of the CNN and the CAE are updated in parallel.

**Algorithm 1** The proposed semi-supervised TL-AMC algorithm for signal classification in ZF-MIMO systems.

**Input:**

- Labeled dataset $\{s_i, l_i\}_{i=1}^{N_C}$
- Unlabeled dataset $\{s_j\}_{j=1}^{N_R}$;
- Leaning rate $\gamma$;
- Batch size $B$;
- Maximum epoches $E$;

Initialize the parameters: $\boldsymbol{\Theta}_F, \boldsymbol{\Theta}_C, \boldsymbol{\Theta}_E, \boldsymbol{\Theta}_D$;
Set $\boldsymbol{\Theta}_F \equiv \boldsymbol{\Theta}_E$;
**while** epoch $e \leq E$ **do**:
  **parafor** each batch **do**:
    **for** each labeled sample in the batch $\{s_i, l_i\}_{i=1}^{B} \subset \{s_i, l_i\}_{i=1}^{N_C}$ **do**:
      Do a feedforward calculation via $f_{CNN}(s_i; \{\boldsymbol{\Theta}_F, \boldsymbol{\Theta}_C\})$;
    **end for**
    Update $\boldsymbol{\Theta}_{CNN} = \{\boldsymbol{\Theta}_F, \boldsymbol{\Theta}_C\}$ by function (13);
  **end parafor**
  **parafor** each batch **do**:
    **for** each unlabeled sample in the batch $\{s_j\}_{j=1}^{B} \subset \{s_j\}_{i=1}^{N_R}$ **do**:
      Do a feedforward calculation via $f_{CAE}(s_j; \{\boldsymbol{\Theta}_E, \boldsymbol{\Theta}_D\})$;
    **end for**
    Update $\boldsymbol{\Theta}_{CAE} = \{\boldsymbol{\Theta}_E, \boldsymbol{\Theta}_D\}$ by function (14);
  **end parafor**
**end while**
**return** $\boldsymbol{\Theta}^* = \{\boldsymbol{\Theta}^*_{CNN}, \boldsymbol{\Theta}^*_{CAE}\}$.

### *E. Comparison Algorithm: Supervised CNN-AMC*

In this paper, for highlighting the performance of TL-AMC, we apply the CNN-based AMC algorithm as a comparison [35], which is trained on labeled dataset, and it is denoted as supervised CNN-AMC. It is noted that the CNN in this algorithm has the same structure and parameters with the CNN module in the TL-AMC for the fair classification performance comparison. In addition, the supervised CNN-AMC is trained on two dataset, and the one dataset is just $\{s_i, l_i\}_{i=1}^{N_C}$, while the other dataset contains large labeled samples, which can be expressed as $\{s_i, l_i\}_{i=1}^{N_C} \cup \{s_j, l_j\}_{j=1}^{N_R}$. Thus, the former supervised CNN-AMC is denoted as "CNN-AMC (Small)" and the latter one is referred as "CNN-AMC (Large)". What's more, the performance of the CNN-AMC, trained on huge labeled samples, can be considered as the upper bound, and the TL-AMC is difficult to go beyond this bond. Our main comparison algorithm is the CNN-AMC trained on the former dataset.

## V. Experiment Results and Discussions

Before showing the performance of TL-AMC, the experimental platform, and the experimental parameters will be first introduced. Firstly, the simulations of neural networks are based on Keras with Tensorflow as backbend, simulation dataset is generated by Matlab, and the device platform is a GPU with one 1080Ti.

Then, we set the batch size $B = 500$, maximum epoches $E = 500$, and learning rate $\gamma = 0.001$. For training and test, we prepare two independent dataset and they contain 20000 and 10000 samples per SNR per type, respectively, and SNR ranges from -10 dB to 10 dB with 2 dB as interval. When training the DRCN in the TL-AMC algorithm, we randomly choose only $N_C$ samples to give them corresponding labels, and the rest $N_R$ samples are unlabeled, where $N_C/(N_C + N_R) \times 100\% = 5\%$ and the labeled samples are very limited.

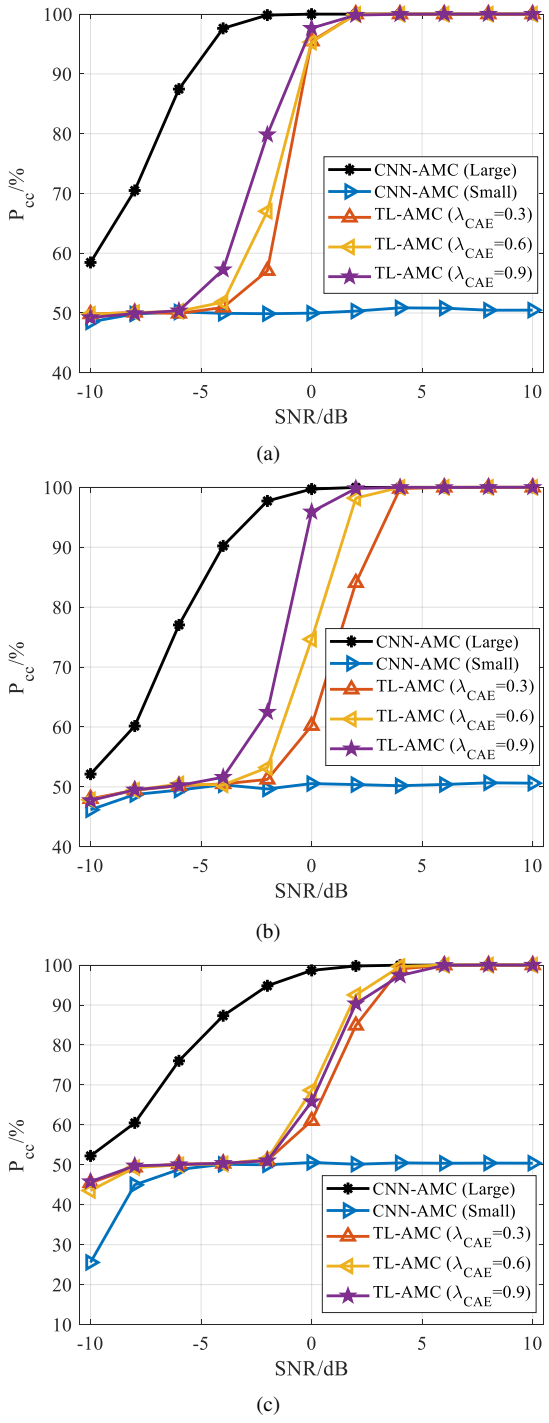


Fig. 6. The classification performances of CNN and TL. (a) $N_r = 4$, $N_t = 1$; (b) $N_r = 4$, $N_t = 2$; (c) $N_r = 4$, $N_t = 4$. It is noted that CNN-AMC (Small) is the extreme condition of TL-AMC with $\lambda_{CAE} = 0$, and $\lambda_{CAE} = 1$.

### *A. Comparing with CNN-AMC (Small) At High SNR*

Confusion matrix (CM), as the main metrics, is applied to compare the performance between TL-AMC and CNN-AMC (Small), because CM can accurately describe the identification

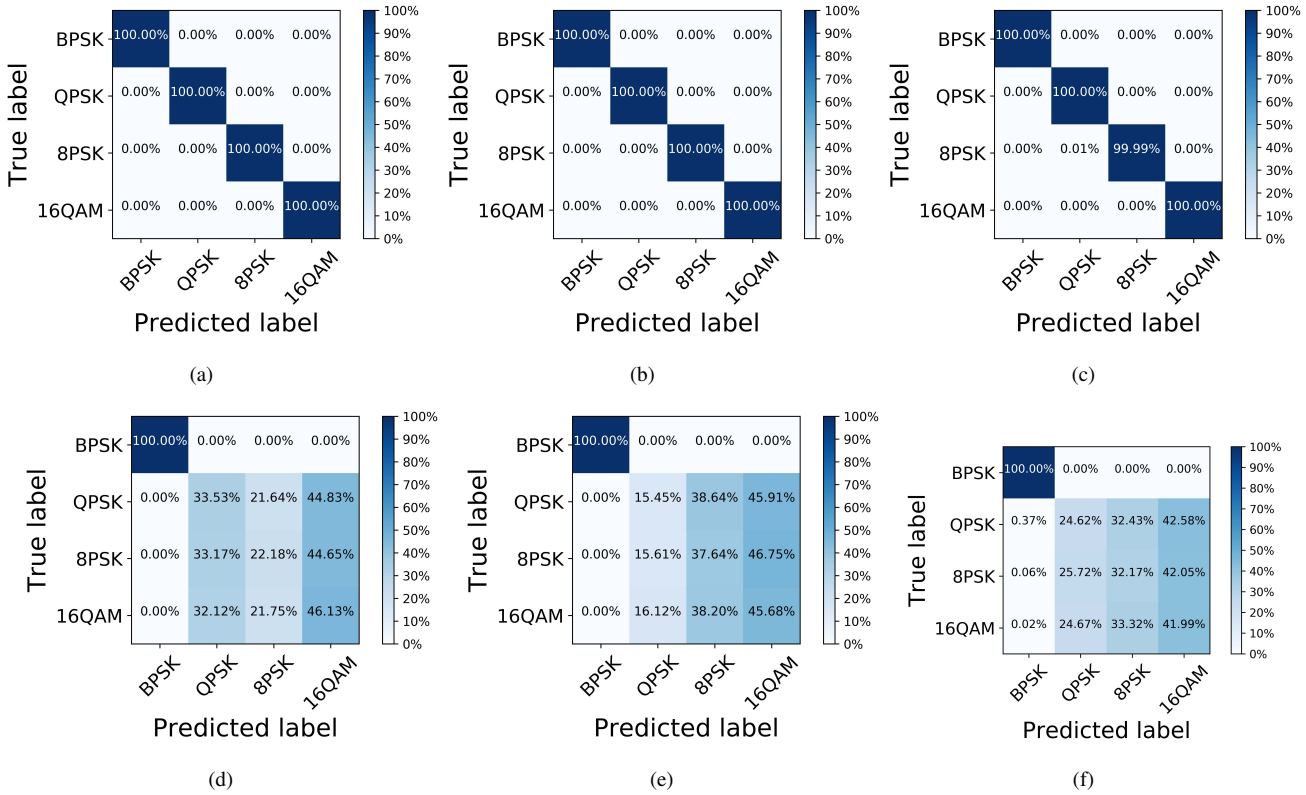


Fig. 4. The confusion matrix of TL-AMC and CNN-AMC (Small) at SNR = 10 dB. (a), (b) and (c) are TL-AMC, and the rest are CNN-AMC. In addition, (a,d), (b,e) and (c,f) are test in ZF-MIMO systems with $\{N_r, N_t\} = \{4, 1\}$, $\{4, 2\}$, and $\{4, 4\}$, respectively. It is obvious that CNN-AMC (Small) has the weak classification capability under few unlabeled samples, while TL-AMC can effectively identify four modulation types at high SNR. In addition, it can be observed that it is easy for BPSK to be identified from other modulation types in either condition, because low order modulation scheme contains little information and is not easy to be confused.

performance of these AMC algorithms for each modulation type. Here, we adopt the CM to reveal which modulation types are hard to be identified in CNN-AMC (Small). CMs of TL-AMC and CNN-AMC (Small) under various ZF-MIMO systems with different antennas are shown in Fig. 4.

From the Fig. 4, it can be observed that CNN-AMC (Small) just has the capability to distinguish BPSK from other modulation types, and it is difficult for CNN-AMC (Small) to classify QPSK, 8PSK and 16QAM in various ZF-MIMO system. However, under the condition of the same number of labeled samples for training, TL-AMC shows it powerful classification performance, and it can perfectly distinguish four modulation types, which has demonstrated the effectiveness of TL-AMC.

Here, we only give the CMs at SNR = 10 dB, because it is obvious that CNN-AMC (Small) is difficult to classify four modulation types even at SNR = 10 dB, and if SNR $<$ 10 dB, the classification performance will be worse.

### *B. Comparing with CNN-AMC (Large)*

In the above section, we has proved the effectiveness of TL-AMC, and here the CCP at each SNR $P_{cc}^{snr}, -10 \leq snr \leq 10$ is applied to indicate the performance gap between the TL-AMC and the CNN-AMC (Large), which can be expressed as

$$P_{cc}^{snr} = \frac{N_{cc}^{snr}}{N_{test}} \times 100\%, -10 \leq snr \leq 10, \tag{15}$$

where $N_{cc}^{snr}$ is the number of correctly classified samples at $snr$ dB; $N_{test}$ is the number of all test samples of {BPSK, QPSK, 8PSK, 16QAM} at each SNR, i, e,. $N_{test} = 40000$

The simulation results in various ZF-MIMO systems are shown in Fig. 6 and Fig. 5. According to these results, we can conclude that TL-AMC has the similar performance with CNN-AMC (Large) under the condition of high SNR, but its performance will decrease sharply, if SNR is too low. In detail, when SNR $\geq$ 0 dB and $\{N_r, N_t\} = \{4, 1\}$ or $\{4, 2\}$, the CCP of TL-AMC is close to that of CNN-AMC (Large), which are also demonstrated by CMs in Fig. 5(a)-5(e).

However, compared with CNN-AMC (Large) in ZF-MIMO system with $\{N_r, N_t\} = \{4, 4\}$, TL-AMC has huge performance gap at SNR = 0 dB. It is also indicated in Fig. 5(c) and Fig. 5(f) that TL-AMC has almost no capability to identify 8PSK and 16QAM, while CNN-AMC (Large) can easily distinguish these two modulation types under $\{N_r, N_t\} = \{4, 4\}$. The same large performance degradation occurs in the ZF-MIMO systems with $\{N_r, N_t\} = \{4, 1\}$ or $\{4, 2\}$, when SNR $<$ 0 dB. In addition, we also provide the curves about the CPP of CNN-AMC (Small) as a comparison. The classification performance of CNN-AMC (Small) just can get up to 50% at most SNRs, which is far weaker than that of TL-AMC.

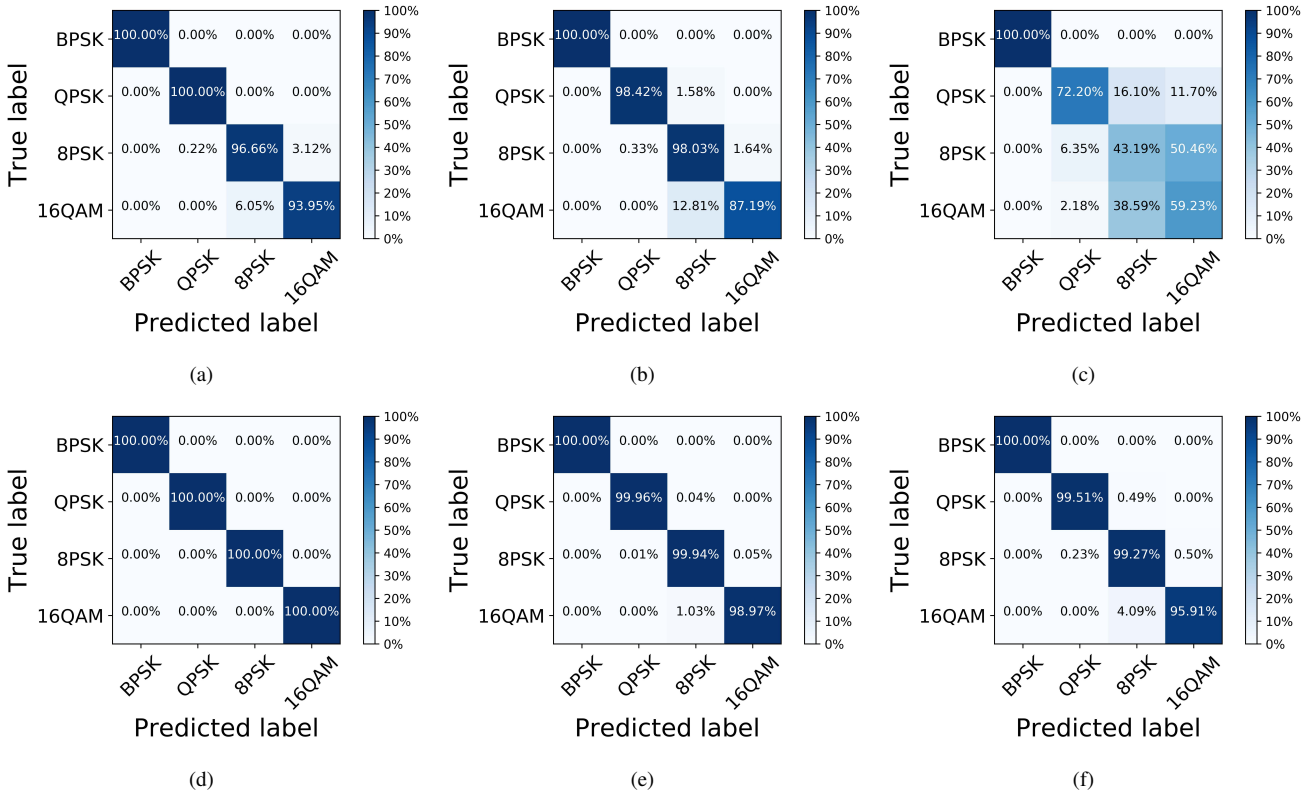


Fig. 5. The confusion matrix of TL-AMC and CNN-AMC (Large) at SNR = 0 dB. (a-c) are TL-AMC, and the rest are CNN-AMC (Large). In addition, (a,d), (b,e) and (c,f) are test in ZF-MIMO systems with $\{N_r, N_t\} = \{4, 1\}$, $\{4, 2\}$, and $\{4, 4\}$, respectively.

What's more, the influence of different $\lambda_{CAE}$ for TL-AMC is also shown in Fig. 6, where $\lambda_{CAE} = \{0.3, 0.6, 0.9\}$ is considered. TL-AMC with a high $\lambda_{CAE}$ can perform well, but it does not means that the higher $\lambda_{CAE}$, the better performance. Too high $\lambda_{CAE}$ can result in the performance decreasing, for example, TL-AMC ($\lambda_{CAE} = 0.9$) has a slightly weaker performance than TL-AMC ($\lambda_{CAE} = 0.6$), when $N_r = 4$, $N_t = 4$. Thus, the choice of $\lambda_{CAE}$ requires many attempts, and the suitable range of $\lambda_{CAE}$ should be $[0.6, 0.9]$.

## VI. Conclusion

In this paper, we propose a TL-AMC method with a small number of labeled samples and a large number of unlabeled samples. In the TL-AMC, a DRCN is proposed and applied, which contains two modules: the CAE for the reconstruction of unlabeled modulation signals and the CNN for the the classification of the labeled modulation signals. In addition, the encoder layer in the CAE has the same structure with that of the feature layer in the CNN, and they share their weights with each other (these two layer can be considered as the same one layer). The reason why we apply the DRCN is that it is well-known that just simple CNN almost has no ability to extract the robust and effective features from a handful of labeled samples, which has been demonstrated by simulation results. Thus, we introduce the unsupervised CAE to make full use of massive unlabeled samples for accurately reconstructing the input samples. Then, the knowledge of the CAE (i.e., its weights) is transferred into the CNN by the weight sharing. The simulation results demonstrate that TL-AMC performs better than CNN-AMC, when there are few labeled samples in the training dataset, and TL-AMC also has the similar performance with CNN-AMC, which is trained on huge labeled samples at high SNR. However, there are still performance gaps between TL-AMC and CNN-AMC, trained on a large number of labeled samples. Hence, our future research will concentrate on few shot learning or other semi-supervised learning model for more effective AMC.

## References

[1] F. Meng, P. Chen, *et al*, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, 2018.

[2] X. Tan, Y. Ueng, Z. Zhang, X. You, and C. Zhang, "A low-complexity massive MIMO detection based on approximate expectation propagation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7260–7272, 2019.

[3] H. Huang, Y. Song, J. Yang, and G. Gui, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[4] Z. Zhu, A. K. Nandi, "Automatic modulation classification: principles, algorithms and applications," *John Wiley and Sons*, 2015.

[5] M. Kulin, T. Kazaz, I. Moerman, and E. D. Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18484–18501, 2018.

[6] W. Zhang, L. Zheng, Y. Xu, G. Wang, and Y. Wu, "Supervised learning method for link adaptation algorithm in coded MIMO-OFDM systems," in *IEEE International Conference on Computer and Communications (ICCC)*, 2018, pp. 414–419.

[7] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.

[8] A. Swami and B. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, 2000.

[9] B. Mao, Z. Md. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? The paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Trans. Comput.*, vol. 66, no. 11, pp. 1946–1960, Nov. 2017.

[10] H. Gacanin, "Autonomous wireless systems with artificial intelligence: A knowledge management perspective," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 51–59, 2019.

[11] B. Mao, F. Tang, Z. Md. Fadlullah, and N. Kato, "An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems," *IEEE Trans. Emerg. Topics Comput.*, in press, doi: 10.1109/TETC.2019.2899407.

[12] J. Sun, W. Shi, Z. Han, J. Yang, and G. Gui, "Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10348–10356, 2019.

[13] X. You, C. Zhang, X. Tan, S. Jin, and H. Wu, "AI for 5G: Research directions and paradigms," *SCIENCE CHINA Information Sciences*, vol. 62, no. 2, pp. 1–13, 2019.

[14] J. Sun, J. Wang, L. Guo, J. Yang, G. Gui, "Adaptive deep learning aided digital predistorter considering dynamic envelope," *IEEE Trans. Veh. Technol.*, in press, doi: 10.1109/TVT.2020.2974506

[15] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari and F. Adachi, "Deep learning for physical-layer 5G wireless techniques: Opportunities, challenges and solutions," *IEEE Wirel. Commun.*, in press, doi: 10.1109/MWC.2019.1900027.

[16] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies towards 6G," *IEEE Wirel. Commun. Mag.*, in press, doi: 10.1109/MNET.001.1900476

[17] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network towards 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[18] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan. 2020.

[19] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 140–150, Jan. 2020.

[20] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: Lightweight automatic modulation classification using deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020.

[21] N. Kato, Z. Md. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inooue, and K. Mizutani, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wirel. Commun. Mag.*, vol. 24, no. 3, pp. 146–153, 2017.

[22] F. Tang, Z. Md. Fadlullah, B. Mao, and N Kato, O. Akashi, T. Inoue, and K. Mizutani, "An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-IoT: A deep learning approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5141–5154, 2018.

[23] Z. Md. Fadlullah, B. Mao, F. Tang, and N. Kato, "Value iteration architecture based deep learning for intelligent routing exploiting heterogeneous computing platforms," *IEEE Trans. Comput.*, vol. 68, no. 6, pp. 939–950, 2019.

[24] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi,"6G: Opening new horizons for integration of comfort, security and intelligence" *IEEE Wireless Commun. Mag.*, in press, doi: 10.1109/MWC.001.1900516.

[25] M. Liu, J. Yang, T. Song, J. Hu, and G. Gui, "Deep learning-inspired message passing algorithm for efficient resource allocation in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 641–653, Jan. 2019.

[26] X. Liu, M. Jia, X. Zhang, W. Lu, "A novel multichannel Internet of things based on dynamic spectrum sharing in 5G communication," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5962–5970, 2018.

[27] X. Liu, and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," *IEEE Transactions on Industrial Informatics*, in press, doi: 10.1109/TII.2019.2947435.

[28] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, in press, doi: 10.1109/JPROC.2019.2957798.

[29] Y. Xu, G. Li, Y. Yang, M. Liu, and G. Gui, "Robust resource allocation and power splitting in SWIPT enabled heterogeneous networks: A robust minimax approach," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10799–10811, Dec. 2019.

[30] C. Zhang, C. Yang, X. Pang, W. Song, W. Xu, S. Zhang, Z. Zhang, and X. You, "Efficient sparse code multiple access decoder based on deterministic message passing algorithm," *IEEE Trans. Veh. Technol.*, in press, doi: 10.1109/TVT.2020.2969020.

[31] 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, 2017.

[32] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, 2019.

[33] S. Hu, Y. Pei, P. Lian and Y. Liang, "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Trans. Veh. Technol.*, in press, doi: 10.1109/TVT.2019.2951594.

[34] Y. Wang, J. Wang, W. Zhang, J. Yang and G. Gui, "Deep learning-based cooperative automatic modulation classification method for MIMO systems," *IEEE Trans. Veh. Technol.*, in press, doi: 10.1109/TVT.2020.2976942.

[35] Y. Wang, J. Gui, Y. Yin, J. Wang, J. Sun, G. Gui, H. Gacanin, H. Sari, and F. Adachi, "Automatic modulation classification for MIMO systems via deep learning and zero-forcing equalization," *IEEE Trans. Veh. Technol.*, in press, doi: 10.1109/TVT.2020.2981995.

[36] M. H. Shah, and X. Dang, "Low-complexity deep learning and RBFN architectures for modulation classification of space-time block-code (STBC)-MIMO systems," *Digital Signal Processing*, in press, doi: 10.1016/j.dsp.2020.102656.

[37] Y. Jiang, D. Wu, Z. Deng, P. Qian, J. Wang, G. Wang, F. Chung, K. Choi, and S. Wang, "Seizure classification from EEG signals using transfer learning, semi-supervised learning and TSK fuzzy system," *IEEE Trans. Neural Systems Rehabil. Eng.*, vol. 25, no. 12, pp. 2270–2284, Dec. 2017.

[38] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 4077–4087.

[39] T. J. O'Shea, N. West, M. Vondal, T. C. Clancy, "Semi-supervised radio signal identification," in *International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 33–38.

[40] Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput. Mater. Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[41] M. Li, O. Li, G. Liu, and C. Zhang, "Generative adversarial networks-based semi-supervised automatic modulation recognition for cognitive radio networks," *Sensors*, vol. 18, no. 11, pp. 3913–3935, 2018.

[42] C. Wang, E. K.S. Au, R. D. Murch, W. H. Mow, R. S. Cheng, and V. Lau, "On the performance of the MIMO zero-forcing receiver in the presence of channel estimation error," *IEEE Trans. Wirel. Commun.*, vol. 6, no. 3, pp. 805–810, 2007.

[43] K. Hassan, I. Dayoub, W. Hamouda, C. N. Nzeza, and M. Berbineau, "Blind digital modulation identification for spatially-correlated MIMO systems," *IEEE Trans. Wirel. Commun.*, vol. 11, no. 2, pp. 683–693, 2011.

[44] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 597–613.

[45] S. Pan, and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[46] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint, arXiv:1502.03167, 2015.