

RESEARCH

Open Access



# Robust automatic modulation classification under noise mismatch

Lan Guo<sup>1</sup>, Rui Gao<sup>1\*</sup> , Yang Cong<sup>1</sup> and Lei Yang<sup>2</sup>

\*Correspondence:  
gaorui@yzu.edu.cn

<sup>1</sup> Department of Information  
Engineering, Yangzhou  
University, Yangzhou, China

<sup>2</sup> Creatcomm Technology Co.,  
Ltd., Suzhou, China

## Abstract

Automatic modulation classification plays a critical role in the intelligent reception of unknown wireless signals. In practice, the dynamic wireless environment brings a great challenge, and the actual test model is inconsistent with the training model. Therefore, aiming at the problem of noise mismatch, this paper proposes a new modulation classification method based on KD-GoogLeNet and Squeeze-Excitation (KD-GSENet). Using the  $k$ -dimensional tree, the complex wireless signals are converted into color images rather than normal constellations, which can enhance the classification features. Considering the attention block has the inherent advantage of assigning more weights to important features, this paper further uses it to improve the GoogLeNet. Finally, extensive experiments are presented including Gaussian noise, non-Gaussian noise, and the scenarios of noise mismatch. Numerical results verify the superior classification performance of the proposed KD-GSENet under different scenarios.

**Keywords:** Automatic modulation classification, Noise mismatch, Deep learning, Constellation, KD-GSENet

## 1 Introduction

Automatic modulation classification (AMC) is an important technique for detecting signal modulation schemes in intelligent communication receivers. As a crucial technique to identify the modulation formats under noise and interference, AMC has been widely used in military, cognitive radio, and crowded electromagnetic spectrum communications [1]. Traditional classification algorithms often require manual extraction by experienced experts. Most traditional methods cannot achieve the requirements of high efficiency and high classification rate.

In recent years, with the rapid development of artificial intelligence, deep learning (DL) has been widely used [2]. The essence of the modulation classification problem is a typical pattern classification problem [3]. The progress of DL promotes the development of modulation classification. Through artificial neural networks, DL can realize automatic feature extraction of different modulated signals. Using DL in AMC can process large amounts of data and extract more comprehensive features without manual feature selection. DL-based AMC method certainly improves classification accuracy; however, most models only consider ideal scenarios such as common Gaussian noise. As the complexity

of the electromagnetic environment increases [4], the background noise often changes dynamically and can be non-Gaussian [5]. In the presence of non-Gaussian noise, the performance of the AMC scheme degrades dramatically. These accelerated the development of AMC. Through preprocessing, the DL-based AMC method can improve the robustness of classification. Therefore, it is necessary to explore the application value of modulation classification in many complex environments [6].

### 1.1 Related works

Modulation classification algorithms are generally categorized into likelihood-based algorithms [7] and feature-based algorithms [8]. Feature-based algorithms typically include three steps: preprocessing, feature extraction, and classification [9]. Moments, cumulants, higher-order statistics, and spectrum are the most commonly used features [9–11].

Machine learning (ML) is an advanced technology that includes different classifiers such as artificial neural networks, K-nearest neighbors, and genetic programming [12]. DL is an important branch of ML, which can simultaneously realize feature extraction and classification. Thus, it is widely used in AMC [13]. An AMC method based on the convolutional neural networks (CNNs) was proposed in [14], which can automatically extract the features and estimate the signal-to-noise ratio (SNR) from sequences. By exploring the interactive features of in-phase/quadrature (I/Q) and amplitude/phase (A/P) [15], Chang et al. proposed a fusion deep neural network. Spatial-temporal characteristics of original complex signals were effectively explored in [16], which helps to obtain more efficient classification features.

Considering that the signal cannot be directly used as the input of the CNN, scholars have proposed different preprocessing methods. A new data preprocessing method was proposed in [17]. Further, many scholars directly converted the signal into a two-dimensional image. O'Shea et al. [13] proposed a modulation classification model based on end-to-end CNN. Inspired by their work, more scholars converted signals into different images. Peng et al. converted complex signals into three-channel constellations and configured models based on GoogLeNet [18]. Zhou et al. [19] proposed a method to classify the received signal without feature extraction, which can automatically learn features from the received signal. The article [20] gave a modulation classification algorithm based on a constellation density matrix to identify different orders of amplitude shift keying (ASK), phase shift keying (PSK), and quadrature amplitude modulation (QAM). Using contrastive full convolutional networks, a novel AMC approach based on a grid constellation matrix was proposed in [21]. In addition to converting signals into constellations, Yan et al. proposed a new feature extraction method based on the cyclic spectrum [22]. They also presented an AMC method for multi-binary QAM signals based on constellation diagram analysis [23]. According to frequency variation with time under different modulation, a short-time discrete Fourier transform was used to convert one-dimensional radio signals into spectral images [24].

For better performance, the researchers designed new networks to extract different representations from the received signals. By adjusting the number of layers and adding new layers, [25] gave an improved AMC network based on CNN. Bu et al. provided a learning architecture that combined adversarial training and knowledge transfer [26].

The long short-term memory (LSTM) was used to learn amplitude and phase information in the time domain [27]. Thien et al. [28] proposed a robust AMC network adopting multiple specific convolution blocks for modern communication systems. They also designed a high-performance CNN structure, which mainly involved multiple high-level processing blocks to learn the intrinsic features of combined waveforms [29]. Huang et al. [30] offered a novel gated recurrent residual neural network. In [31], residual networks were used to extract discriminant features. Aiming at the classification accuracy and calculation time, the article [32] introduced an efficient AMC scheme by exploiting the bottleneck structure of the residual network. Using CNN and gate recurrent unit as feature extraction layers, [33] presented an efficient model based on phase parameter estimation and transformation.

In actual communication, signal modulation classification is vulnerable to dynamic environments. Therefore, it is crucial to come up with more robust AMC methods for different cases. To improve the classification performance under impulsive noise, Zhang et al. pointed out a modulation classification method based on the cyclic correlation entropy spectrum [1]. The paper [34] generated the feature vector of missing modulated signals based on semantic feature vector, which greatly improved the classification accuracy of undiscovered classes. Adopting the Cauchy distribution function as a robust feature of acoustic noise, an improved constellation was presented [35]. To overcome the intra-class classification problem caused by the dynamic changes, Luan et al. [36] suggested an AMC method based on the multi-scale network.

Thus, it is urgent to find a method that can adapt to different noise environments.

## 1.2 Contributions

In this paper, a new AMC method is proposed. By combining preprocessing and improving the network, the proposed method improves the classification accuracy. The method has good robustness to non-Gaussian noise and noise mismatch. The contributions of this paper are summarized as follows:

- A GoogLeNet and Squeeze-Excitation (GSENet) network is proposed. By assigning more weight on important features, the network combines a self-attention mechanism to enhance the discrimination and expression.
- A method of  $k$ -dimensional tree (KD-tree) preprocessing is further introduced, which directly converts the signals into three-channel constellations. Unlike traditional constellations, it enlarges the differences between different modulations by combining more characteristics of signals. The KD-GoogLeNet and Squeeze-Excitation (KD-GSENet) is capable of identifying the received signals in the case of noise mismatch.
- Extensive numerical results are performed to evaluate the performance of the KD-GSENet under Gaussian noise, non-Gaussian noise, and the case of noise mismatch. Besides, the classification accuracy and computational complexity are compared with other methods in this article. Numerical results verify that the proposed method not only has superior classification accuracy under Gaussian noise but also has little performance loss under non-Gaussian noise. Moreover, compared with other methods,

the proposed has high robustness and generalization in the case of noise mismatch, while the increase of algorithm complexity is not significant.

### 1.3 Organization

The rest of this paper is organized as follows. In Sect. 2, the system model is summarized. Section 3 presents the proposed AMC method. The numerical results are presented and discussed in Sect. 4. Section 5 draws conclusions.

## 2 System model

In this section, the signal model and the model of background noise are briefly introduced.

### 2.1 Signal model

This paper aims to identify the correct modulation scheme among binary phase shift keying (BPSK), four amplitude shift keying (4ASK), quadrature phase shift keying (QPSK), offset QPSK (OQPSK), eight phase shift keying (8PSK), 16-ary quadrature amplitude modulation (16QAM), 32-ary quadrature amplitude modulation (32QAM), and 64-ary quadrature amplitude modulation (64QAM). According to the traditional modulation classification model [1], in which the receiver is equipped with a single antenna, the received signal can be represented as

$$y(n) = hs(n) + w(n), \quad n = 1, 2, \dots, N, \quad (1)$$

where  $y(n)$  is the received signal,  $h$  is the channel gain, which is invariant during the classification process,  $s(n)$  is the transmitted signal with eight possibilities.  $N$  is the sample number,  $w(n)$  is the generalized Gaussian noise (GGN) with zero mean, which will be discussed in the posterior subsection.

### 2.2 Model of background noise

In addition to Gaussian noise, non-Gaussian noise [37] is considered in this paper. Non-Gaussian noise is a random process in which the probability density function (PDF) does not satisfy the Gaussian distribution. GGN includes Gaussian noise and partially non-Gaussian noise, the PDF of GGN is

$$f_{\gamma}(\gamma) = \frac{\beta}{2\nu\Gamma(1/\beta)} \exp\left(-\left|\frac{\gamma - \varpi}{\nu}\right|^{\beta}\right), \quad (2)$$

where  $\varpi = 0$  is the mean,  $\nu$  is the “scale parameter”,  $\beta$  is the “shape parameter”, and  $\Gamma(\cdot)$  denotes the Gamma function. In particular, equation (2) represents the Gaussian distribution when  $\beta = 2$ , which is Gaussian noise. The remaining cases are part of non-Gaussian noise, such as the Laplacian distribution when  $\beta = 1$ . In GGN, the expectation and variance of noise are given in [38] as

$$\mathbb{E}[\gamma(n)] = \varpi = 0, \quad (3)$$

$$\mathbb{D}[\gamma(n)] = \vartheta_\gamma^2 = \frac{v^2 \Gamma(3/\beta)}{\Gamma(1/\beta)}, \quad (4)$$

where  $\mathbb{E}[\cdot]$  and  $\mathbb{D}[\cdot]$  denote the expectation and variance operators. Thus, the SNR  $\varphi$  can be expressed as

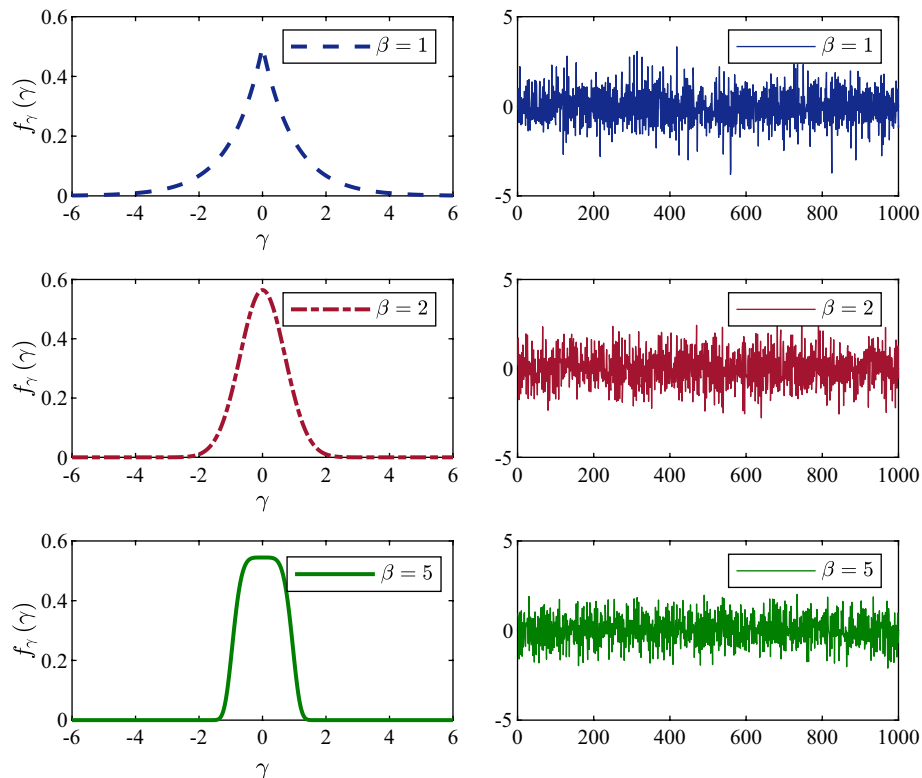
$$\varphi = \frac{P}{\frac{v^2 \Gamma(3/\beta)}{\Gamma(1/\beta)}} = \frac{P \Gamma(1/\beta)}{v^2 \Gamma(3/\beta)}, \quad (5)$$

where the signal power  $P = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N |s(n)|^2$ . Some possible shapes and realizations of generalized Gaussian distribution with the same variance are shown in Fig. 1. It can be noticed that, the Laplacian time series ( $\beta = 1$ ) exhibits more spikes or outliers than time series with larger  $\beta$ .

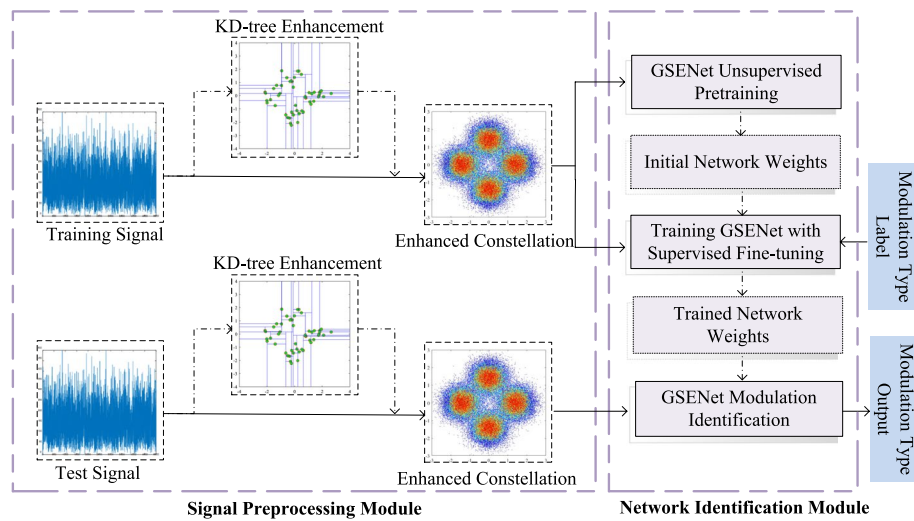
In real communication, the environment is variable. Not only Gaussian noise and non-Gaussian noise, but also the cases of noise mismatch are considered in the following experiments. Noise mismatch in this paper refers to the noise inconsistency in training and testing, in which the test data is determined by the real wireless environment.

### 3 Proposed AMC method

In this paper, an AMC method based on KD-tree enhancement and GSENet is proposed. Figure 2 is the overall structure, which mainly includes a signal preprocessing module and a network identification module.



**Fig. 1** Some shapes and realizations of generalized Gaussian distribution with different  $\beta$



**Fig. 2** The schematic diagram of the KD-GSENet

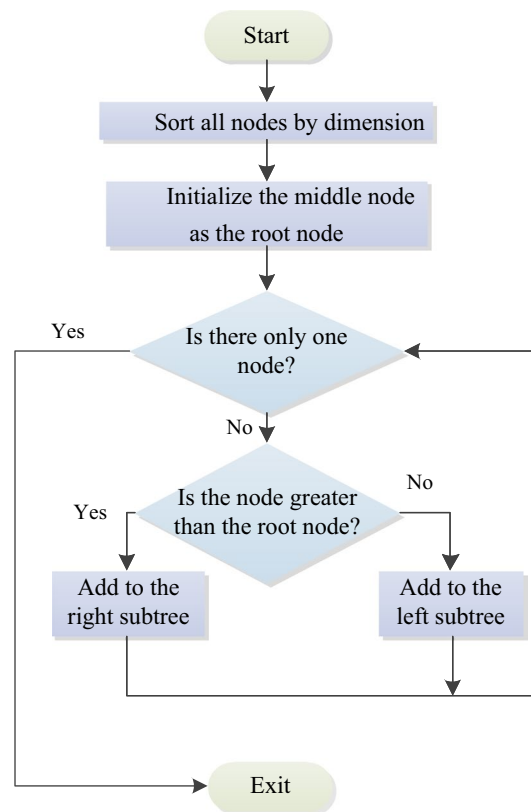
First, in the signal preprocessing module, according to the difference in distance between each signal point, the KD-tree enhancement strategy is used to color different modulation types. The generated signals are directly converted into color constellations. Secondly, in the network identification module, the enhanced constellation is used to train the network for modulation classification. The key is to build the GSENet model, which introduces the Squeeze-and-Excitation (SE) block in the sub-module and auxiliary classifier. It improves the ability to classify different signals. Further, a batch normalization (BN) layer is added and the activation function is updated to the rectified linear unit (ReLU) [39], which effectively enhances the generalization of the network. Finally, the trained KD-GSENet is used to identify the enhanced constellations under different  $E_b/N_0$ . The specific algorithm is as follows.

### 3.1 Preprocessing: KD-tree enhancement

The enhanced constellation is drawn by the method of the KD-tree [40] neighborhood point search. To improve the classification characteristics of images, the radio signals are directly converted into color constellations. As a node of the KD-tree, each signal point is divided into a root node or a leaf node.

For a sample set composed of  $n$   $d$ -dimensional data, the eigenvalues of any sample can be used as the root node. To ensure the fastest search to the nearest point, the construction of a balanced binary tree is shown in Fig. 3.

1. Determine the root node Select dimensions according to the sequential traversal method, and all nodes are sorted by the division dimension. Initially, the intermediate node is used as the root node.
2. Determine left and right subtrees Compare the value of one node with a split node of the same dimension. When the value of a node is greater than the split node, it should be placed in the subtree to the right of the split node. Conversely, if the value of a node is less than the split node, it will be placed in the left subtree.



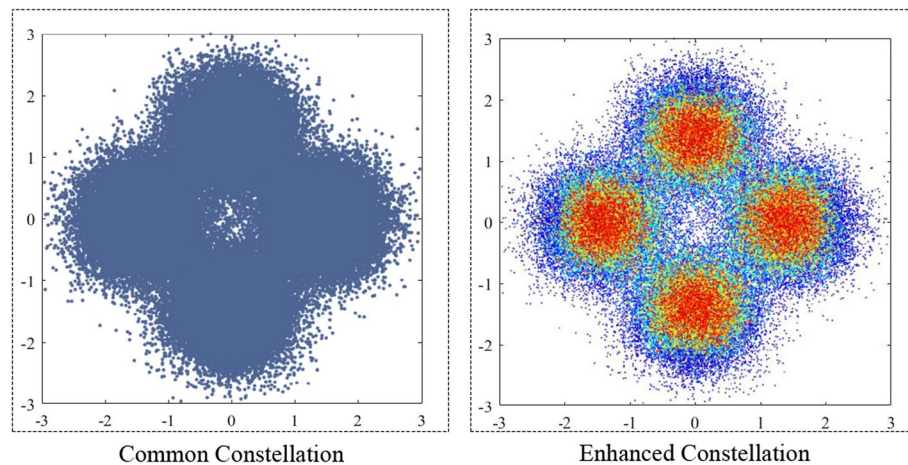
**Fig. 3** Build a balanced KD-tree

- Recursive process According to the tree building in step 2, all nodes are integrated into one tree by recursing the left and right subtrees.

According to the constructed KD-tree, the nearest neighbor points are searched. First, assuming that the “current nearest neighbor” is its parent node, the minimum distance is the distance to the parent node. During the backtracking, if the distance between the child node and the target node is smaller than the distance between the “current nearest neighbor” and the target node, the “current nearest neighbor” is updated to the selected child node. The iteration is terminated until coming back to the root of the tree. The minimum distance between the target node and its nearest neighbor is calculated. After obtaining the nearest distance of each signal point, it is approximated as the density of that position. Finally, all signal points are colored according to density.

The enhanced constellation of QPSK is shown in Fig. 4. It can be seen that each point in the preprocessed constellation has different information. No longer independent or have equal information. This processing method condenses more time-accumulated features of received signals in the constellation, which enhances its separability and achieves feature enhancement. Thus, the received signals have been converted into color images with dimensions of  $3 \times 224 \times 224$ .





**Fig. 4** Constellation enhancement contrast

### 3.2 Network optimization

The DL model consists of multiple layers, each containing multiple neurons for automatic feature extraction. The initial layer extracts abstract features. Deep layers obtain important features by applying multiple nonlinear transformations on the output of the previous layer. GoogLeNet [41] is a DL structure proposed by the Google team, which won the ImageNet competition with a significant advantage. The parallel structure adopted by the model can integrate feature information of different scales, the model also uses  $1 \times 1$  convolution kernels for dimensionality reduction and mapping, besides, two auxiliary classifiers are added to help with training. By replacing the traditional dropout fully connected layer with the average pooling layer, the parameters of the model are greatly reduced. Therefore, to improve the classification ability of modulated signals, the GoogLeNet is introduced.

To speed up the training of the network, a BN layer is added after each convolutional layer. Convolutional layers extract image features. By normalizing the same feature of different samples, the BN layer is used to normalize the feature data, which accelerates the network training speed and improves the generalization ability. ReLU is used as the activation function.

Original signals have been converted into color constellations by the transformation of KD-tree enhancement. Considering that the SE block can assign more weights to important features [42], it is used for feature classification on the sub-modules and auxiliary classifiers of GoogLeNet. Specifically, the SE module performs adaptive average pooling on each channel according to the obtained feature matrix. The output vector is obtained through two fully connected layers. The number of nodes in the first fully connected layer is  $1/4$  of the characteristic matrix channel. The number of nodes in the second fully connected layer is consistent with the input characteristic matrix channel. The vector output of the second fully connected layer analyzes the weight relation of each channel. Important channels are given larger weights, while unimportant channels correspond to smaller weights. Each result is multiplied by the corresponding number of channels, assigning more weight to important features.



Figure 5 shows the specific sub-module. The first sub-module is elaborated as an illustration. The feature matrix  $\mathcal{Y} \in \mathbb{R}^{H \times W \times C}$  output from the previous layer is input into four branches  $Y_k, k = \{1, 2, 3, 4\}$  for processing

$$\begin{aligned} Y_1 &= \mathcal{C}(\mathcal{Y}), \\ Y_2 &= \mathcal{C}(\mathcal{C}(\mathcal{Y})), \\ Y_3 &= \mathcal{C}(\mathcal{C}(\mathcal{Y})), \\ Y_4 &= \mathcal{P}(\mathcal{C}(\mathcal{Y})), \end{aligned} \quad (6)$$

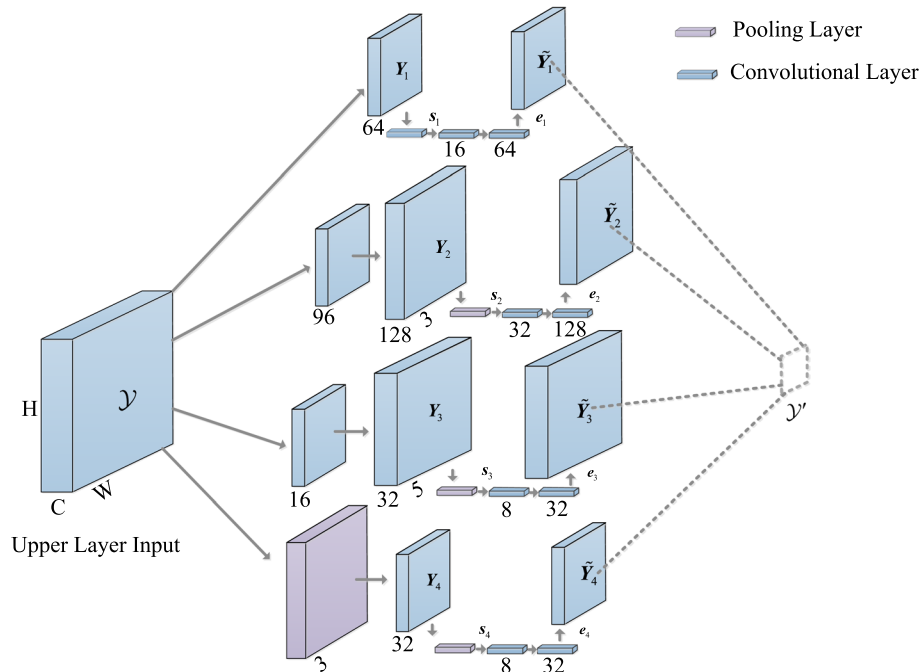
The first branch is a convolutional layer  $\mathcal{C}$  with a kernel size of  $1 \times 1$ . The second branch adopts a  $1 \times 1$  convolutional layer with a dimensionality reduction function and a  $3 \times 3$  convolutional layer. The third branch passes through a  $1 \times 1$  convolutional layer with a dimensionality reduction function and a  $5 \times 5$  convolutional layer. The fourth branch passes through a  $3 \times 3$  maximum pooling layer  $\mathcal{P}$  and a  $1 \times 1$  convolutional layer for dimensionality reduction. In each branch, the parameter of the  $c$ th filter is  $y_{k,c} \in \mathbb{R}^{H \times W \times K}$ . The output is  $Y_k = [y_{k,1}, y_{k,2}, \dots, y_{k,C}]$ . A statistic  $s_{k,c} \in \mathbb{R}^{C \times K}$  is generated by shrinking the spatial dimensions  $H \times W$  of  $Y_k$ , the  $c$ th element of  $s_k$  is calculated by

$$s_{k,c} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W y_{k,c}(i, j). \quad (7)$$

After obtaining  $s_k$ , there are two fully connected layers

$$e_k = \sigma(g(s_k, W)) = \sigma(W_2 \delta(W_1 s_k)), \quad (8)$$

where  $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ ,  $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ ,  $r$  is the reduction ratio used to reduce the dimension of the fully connected layer.  $\delta$  represents the sigmoid activation function,  $\sigma$  refers



**Fig. 5** The structure of the improved sub-module

to the ReLU, which is used as the activation function to introduce nonlinearity into the network. In addition, ReLU is beneficial to avoid gradient vanishing and explosion. It increases the sparsity of the network and alleviates the over-fitting problem. The expression of ReLU is

$$f(z) = \max(0, z), \quad (9)$$

where  $z$  is the neuron. Note that the convolutional layer is used to replace the pooling layer when implemented. Since the down-sampling during pooling may confuse the information in the enhanced constellation. In addition, successive convolutional layers can improve the nonlinearity of the network and limit the scale, which helps to enhance learning and prevent overfitting.

The final output of the block is obtained by rescaling  $Y_k$  with the activation  $e_k$

$$\tilde{y}_{k,c} = e_{k,c} y_{k,c}, \quad (10)$$

where  $\mathbf{Q}_k = [\tilde{y}_{k,1}, \tilde{y}_{k,2}, \dots, \tilde{y}_{k,C}]$ ,  $e_{k,c} y_{k,c}$  refers to channel-wise multiplication between the scalar  $e_{k,c}$  and the feature map  $y_{k,c} \in \mathbb{R}^{H \times W \times K}$ .

The enhanced feature matrices of different scales are obtained through four branches  $Y_k, k = \{1, 2, 3, 4\}$ . Each feature matrix has the same height and width. After processing, the obtained four feature matrices are spliced by depth. Thus, the output feature matrix is

$$\mathcal{Y}' = \text{concat}(\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2, \tilde{\mathbf{Y}}_3, \tilde{\mathbf{Y}}_4), \quad (11)$$

These four branches increase the width of the network, enabling GSENet to learn multi-scale information. Using the concat operation [29], the features of different convolutional layers are merged, which increases the non-linear capacity of the trainable network.

The structure of the improved auxiliary classifier is shown in Fig. 6. First, the average pooling layer has a  $5 \times 5$  pooling kernel and a 3 stride. 128 convolutional layers with  $1 \times 1$  convolution kernels are used to reduce the dimension. The weights of different channels are obtained through the SE block and multiplied by the corresponding channels. The specific calculation process is the same as the sub-module. The obtained feature matrix is flattened. To reduce over-fitting, the dropout function at 50% is used to randomly inactivate neurons during forward propagation. The number of nodes in the output layer is the same as the modulation types. Finally, the output is converted to the probability that the input signal belongs to each candidate modulation format through the Softmax [43], which can be expressed as

$$p_i = \text{Softmax}(y_i) = \frac{e^{y_i}}{\sum_{c=1}^M e^{y_c}}, \quad (12)$$

where  $y_i$  is the output value of the  $i$ th neuron,  $M$  is the number of output neurons equal to the number of modulation types. The output value of the multi-classification can be converted into a probability distribution in the range  $[0, 1]$  and 1 by Softmax,  $p_i$  is the probability of the corresponding neuron.

In training, 8 modulated signals need to be identified. After calculating the probability of each type through Softmax, the loss function is used to find the optimal weight parameter, the formula is

$$\text{Loss}_i = -\log p_i = -\log \frac{e^{y_i}}{\sum_{c=1}^M e^{y_c}} = -\left(y_i - \log \sum_{c=1}^M e^{y_c}\right), \quad (13)$$

the probability models with smaller cross entropy are closer.

Using adaptive moment estimation (Adam) [44] as the optimizer, different adaptive learning rates are assigned to different weight parameters. By updating the model parameters, the loss function is minimized. The Adam algorithm combines momentum and adaptation to avoid the cold start problem. Unlike stochastic gradient descent, Adam uses the gradient second moment to accelerate the convergence speed. On this basis, the back-propagation algorithm is introduced to update the weights until the loss converges to a stable value.

The KD-GSENet algorithm is summarized in Algorithm 1.

---

**Algorithm 1** The KD-GSENet Algorithm

---

**Input:** The modulated signal dataset and its label  $L$ , actual received signal  $y(n)$ ;

**Output:** The modulation type  $l$  of  $y(n)$ .

---

- 1: Begin the training stage.
  - 2: Transform, scale, and normalize the training signal.
  - 3: Receive the constellation set  $\mathcal{Y}_1$  according to Fig. 3.
  - 4: Build a KD-GSENet model based on GoogLeNet, Fig. 5, and Fig. 6.
  - 5: Use  $\mathcal{Y}_1$ , achieve the pre-training weight matrix  $Q$  and the bias value  $b$ .
  - 6: **while** (13) not convergence **do**
  - 7:   Update  $Q$  and  $b$ .
  - 8: **end while**
  - 9: Begin the test stage.
  - 10: Transform, scale, and normalize the test signal.
  - 11: Receive the dataset  $\mathcal{Y}_2$  to be identified.
  - 12: Use the trained KD-GSENet to extract features of  $\mathcal{Y}_2$ , obtain  $l$ .
- 

#### 4 Numerical results and discussion

In this section, numerical results are performed to verify the superiority and robustness of the proposed method. In the experiment, the KD-GSENet classification method is verified by the simulation dataset containing 8 modulated signals. The proposed method is further compared with other classification methods. At the same time, the influence of KD-tree enhancement, the different noise, and the impact of  $E_b/N_0$  changes in the classification performance are also analyzed through experiments. Finally, the implementation complexity and processing speed of these methods are compared.

The experiment is measured by the ratio of bit energy to noise power spectral density ( $E_b/N_0$ ), which is classically defined as the ratio of energy per bit ( $E_b$ ) to the

spectral noise density ( $N_0$ ) [45–47].  $E_b/N_0$  is a measure of the normalized performance for the entire communication system.

#### 4.1 Parameters setting

The experiment selects 8 commonly used modulated signals: BPSK, 4ASK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, and 64QAM. There are 4800 training samples (600 for each modulation method), 1600 verification samples (200 for each modulation method), and 1600 test samples for each  $E_b/N_0$  (200 for each modulation method). Without loss of generality, the symbol rate is set to 1MHz and the sampling rate is 8 MHz. The default background environment is GGN. For comprehensive coverage, the training signals are randomly distributed with  $E_b/N_0 = [-7, 5]$  dB. To evaluate the classification performance of the algorithm with  $E_b/N_0$  changes, the test signals are  $-7$  dB,  $-6$  dB,  $-5$  dB,  $-4$  dB,  $-3$  dB,  $-2$  dB,  $-1$  dB,  $0$  dB,  $1$  dB, and  $2$  dB.

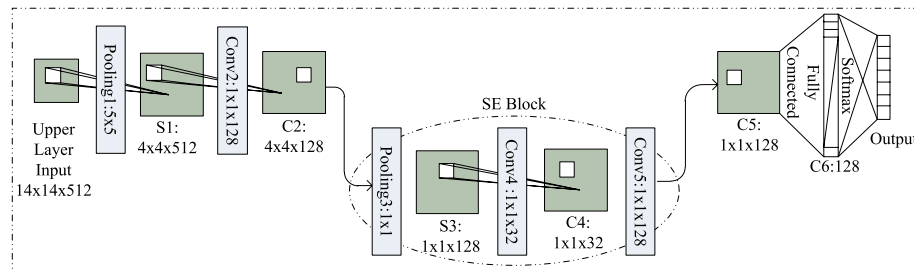
The neural network model is built under PyTorch. The graphics card of the experimental computer is *NVIDIA GeForce MX350*, the processor is *Intel(R) Core(TM) i7-1065G7 CPU@1.30GHz/1.50GHz*. The basic network of KD-GSENet is built on GoogLeNet. The structures of the improved sub-module and auxiliary classifier are shown in Figs. 5 and 6. The specific parameters are shown in Tables 1 and 2. In the table, the second column represents the output size, the last digit is the number of channels. In the third column,  $w \times h \times c$  represents the

**Table 1** The parameters of auxiliary classifier

Layer name	Output size	Layer parameters
<i>Branch 1</i>		
Convolutional Layer	$28 \times 28 \times 64$	$1 \times 1 \times 64$ Convolution+ReLU6
Pooling Layer	$1 \times 1 \times 64$	$1 \times 1$ Adaptive batch normalization
Convolutional Layer	$1 \times 1 \times 16$	$1 \times 1 \times 16$ Convolution+ReLU
Convolutional Layer	$1 \times 1 \times 64$	$1 \times 1 \times 64$ Convolution+Hard Sigmoid
<i>Branch 2</i>		
Convolutional Layer	$28 \times 28 \times 96$	$1 \times 1 \times 96$ Convolution+ReLU6
Convolutional Layer	$28 \times 28 \times 128$	$3 \times 3 \times 128$ Convolution+ReLU6
Pooling Layer	$1 \times 1 \times 128$	$1 \times 1$ Adaptive batch normalization
Convolutional Layer	$1 \times 1 \times 32$	$1 \times 1 \times 32$ Convolution+ReLU
Convolutional Layer	$1 \times 1 \times 128$	$1 \times 1 \times 128$ Convolution+Hard Sigmoid
<i>Branch 3</i>		
Convolutional Layer	$28 \times 28 \times 16$	$1 \times 1 \times 16$ Convolution+ReLU6
Convolutional Layer	$28 \times 28 \times 32$	$5 \times 5 \times 32$ Convolution+ReLU6
Pooling Layer	$1 \times 1 \times 32$	$1 \times 1$ Adaptive batch normalization
Convolutional Layer	$1 \times 1 \times 8$	$1 \times 1 \times 8$ Convolution+ReLU
Convolutional Layer	$1 \times 1 \times 32$	$1 \times 1 \times 32$ Convolution+Hard Sigmoid
<i>Branch 4</i>		
Pooling Layer	$28 \times 28 \times 192$	$3 \times 3$ Max pooling
Convolutional Layer	$28 \times 28 \times 32$	$1 \times 1 \times 32$ Convolution+ReLU6
Pooling Layer	$1 \times 1 \times 32$	$1 \times 1$ Adaptive batch normalization
Convolutional Layer	$1 \times 1 \times 8$	$1 \times 1 \times 8$ Convolution+ReLU
Convolutional Layer	$1 \times 1 \times 32$	$1 \times 1 \times 32$ Convolution+Hard Sigmoid

**Table 2** The parameters of sub-module

Layer name	Output size	Layer parameters
Pooling layer	$4 \times 4 \times 512$	$5 \times 5$ Average pooling
Convolutional layer	$4 \times 4 \times 128$	$1 \times 1 \times 128$ Convolution + batch normalization + ReLU6
Pooling layer	$1 \times 1 \times 128$	$1 \times 1$ Adaptive batch normalization
Convolutional layer	$1 \times 1 \times 32$	$1 \times 1 \times 32$ Convolution + ReLU
Convolutional layer	$1 \times 1 \times 128$	$1 \times 1 \times 128$ Convolution + Hard Sigmoid
Flatten layer	128	feature flattening
Fully connected layer	8	8 Fully connected + Softmax

**Fig. 6** The structure of the improved auxiliary classifier

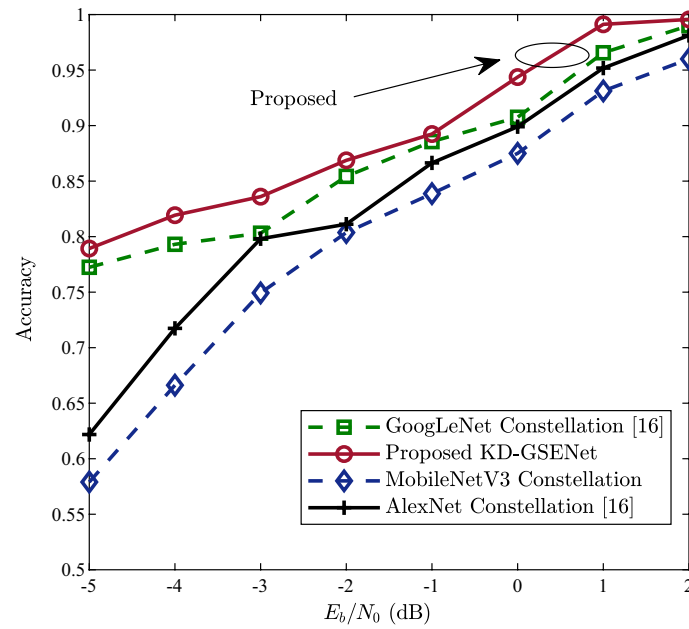
convolution parameter. The number of output channels is  $c$ ,  $w \times h$  is the size of the convolution kernel. During training, an optimization algorithm with a learning rate  $\mu = 0.0003$  is adopted.

#### 4.2 Numerical results and discussion

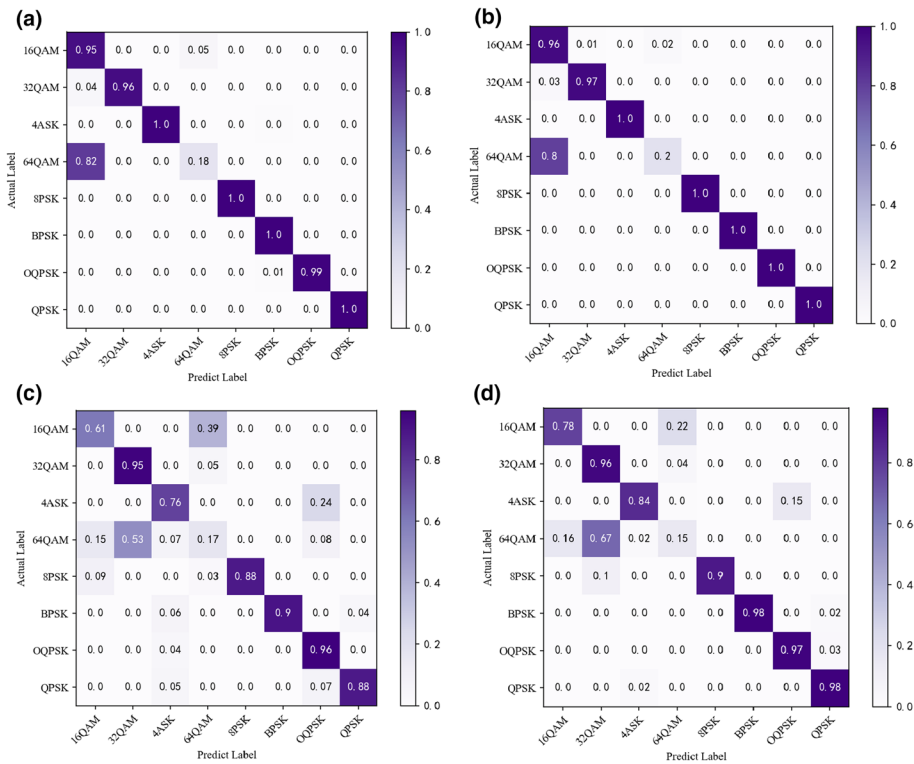
Figure 7 shows the classification performance of four methods under different  $E_b/N_0$ . The classification accuracy is obtained by averaging the classification performance of 8 modulation types. The proposed method is also compared with the modulation classification method based on AlexNet [48], MobileNetV3 [49], and GoogLeNet [18]. As shown in Fig. 7, the modulation classification performance improves with  $E_b/N_0$  increasing for all algorithms. Obviously, the proposed KD-GSENet is superior to other models, achieving higher accuracy at the same  $E_b/N_0$ .

To further evaluate the effect of KD-tree enhancement on various modulated signals, the confusion matrices are drawn. The first two subgraphs of Fig. 8 show the confusion matrices of the GSENet model and the proposed KD-GSENet model with  $E_b/N_0 = -1$  dB. The classification accuracy of the two models is 88.5% and 89.2%, respectively. It can be seen that using the KD-GSENet model achieves the 100% classification of 4ASK, BPSK, QPSK, 8PSK, and OQPSK. The method of KD-tree enhancement improves the classification accuracy of both 8 signals and the intra-class. The confusion matrices with  $E_b/N_0 = -4$  dB present similar results in the last two subgraphs of Fig. 8.

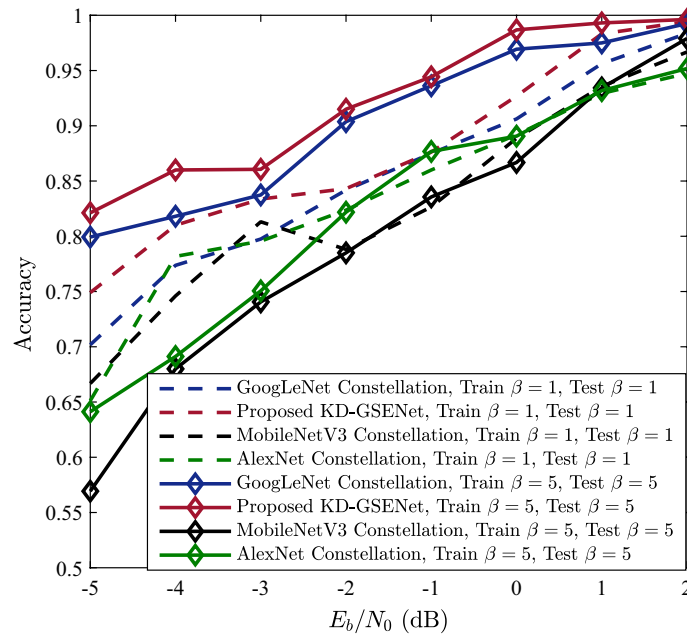
Figure 9 shows the average classification accuracy under non-Gaussian noise. Two cases are considered, respectively, for training  $\beta = 1$  corresponding to test  $\beta = 1$  and training  $\beta = 5$  corresponding to test  $\beta = 5$ . In different noise environments, the classification accuracy is improved by using KD-GSENet. The classification accuracy generally improves with the increase of  $E_b/N_0$ .



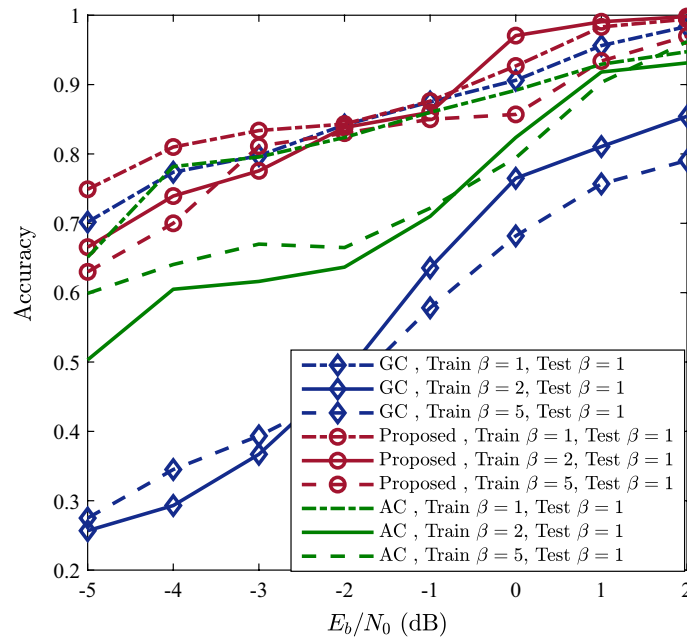
**Fig. 7** The classification performance of four methods under different  $E_b/N_0$



**Fig. 8** Confusion matrices of GSENet model and KD-GSNet model. **a** GSENet with  $E_b/N_0 = -1$  dB. **b** KD-GSNet with  $E_b/N_0 = -1$  dB. **c** GSENet with  $E_b/N_0 = -4$  dB. **d** KD-GSNet with  $E_b/N_0 = -4$  dB



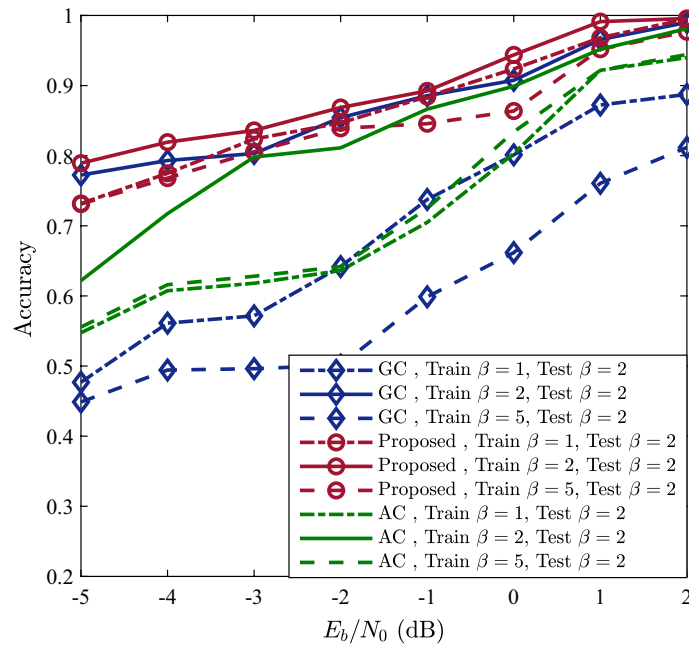
**Fig. 9** The average classification accuracy under non-Gaussian noise



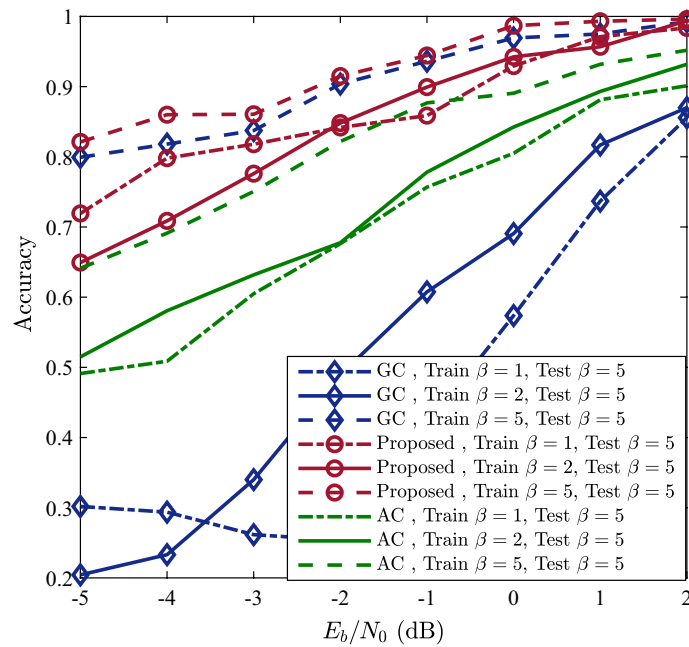
**Fig. 10** The classification accuracy when the training set and test set do not match (Test  $\beta = 1$ )

In practice, the case of noise mismatch is easy to occur. Therefore, some experiments were done for this common situation. Figures 10, 11, and 12 show the classification accuracies when the training and test sets do not match. Three methods are considered: GoogLeNet constellation (GC), AlexNet constellation (AC), and KD-GSE Net.





**Fig. 11** The classification accuracy when the training set and test set do not match (Test  $\beta = 2$ )



**Fig. 12** The classification accuracy when the training set and test set do not match (Test  $\beta = 5$ )

**Remark 1** Obviously, in the case of noise mismatch, the proposed KD-GSENet shows little performance loss while the classification accuracies of other methods drop severely. Not only do the experimental results demonstrate the better performance of

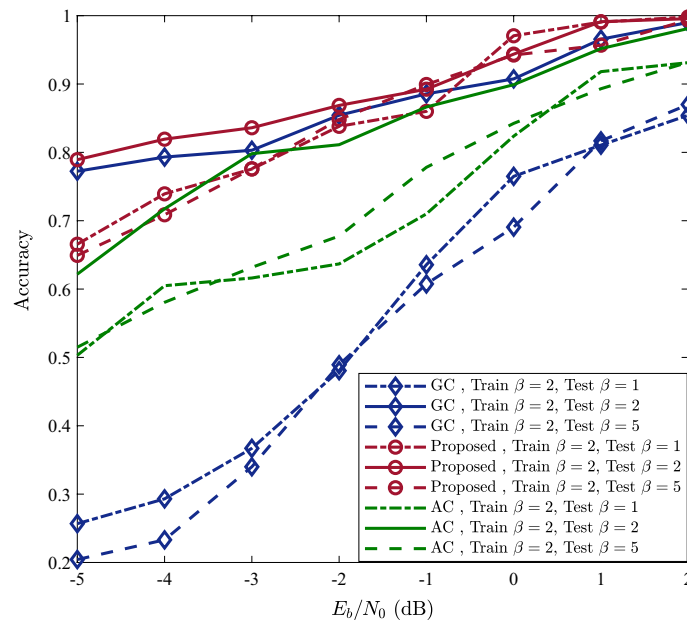
the proposed method under Gaussian noise and non-Gaussian noise, but also reveal its robustness under noise mismatch.

Figure 13 shows the classification accuracy curve under noise mismatch. The figure shows the case of Gaussian noise in training and non-Gaussian noise in test. It can be seen that the classification accuracies of the three methods are similar when both training and test are Gaussian noise (Train  $\beta = 2$ , Test  $\beta$ ), and the proposed method is slightly higher than the others. When the noise is mismatched, the classification accuracy of the proposed method decreases slightly, while the other two methods decrease significantly. This result shows the robustness of the proposed method under noise mismatch, which is consistent with the conclusion of the previous experiment.

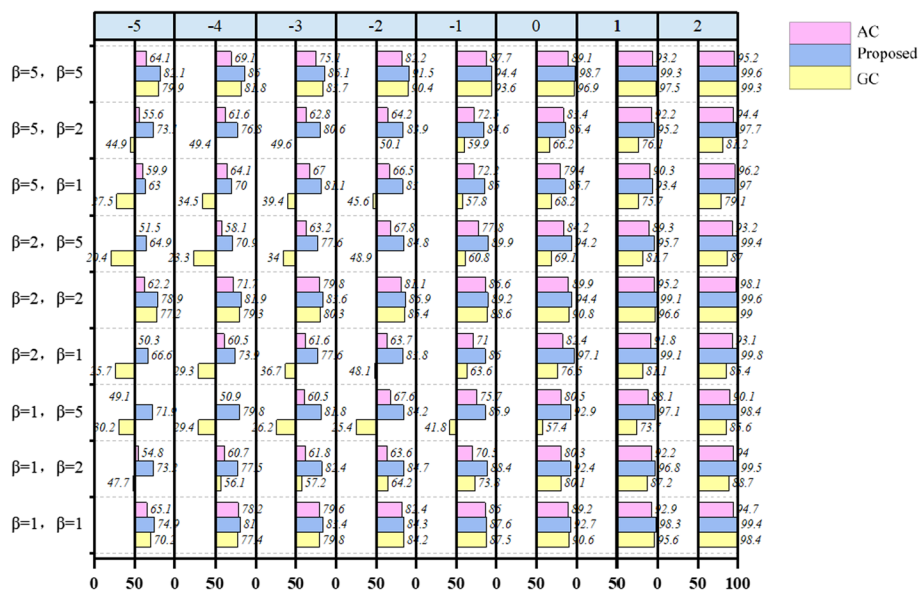
Figure 14 shows the box plots under noise mismatch, which can more intuitively compare changes in results. Box plots more visually represent the variability of results. In the figure, the pink, blue, and yellow boxes represent the three modulation classification methods of AC, the proposed method, and GC, respectively. Intuitively, the method proposed in this paper (blue box) has the best classification accuracy in different noise mismatch scenarios, which shows that the proposed method has superior robustness in noise mismatch.

#### 4.3 Algorithm complexity analysis

As shown in Table 3, the total parameter size and parameter storage size of the proposed method are smaller than AC, since the average pooling layer is adopted and the fully connected layer in AC is abandoned. The proposed takes up more memory than GC because it introduces SE blocks, which increase the number of parameters. Mobile-NetV3 constellation (MC) has the lightest network due to the depth-wise separable convolution and inverted residual structure.

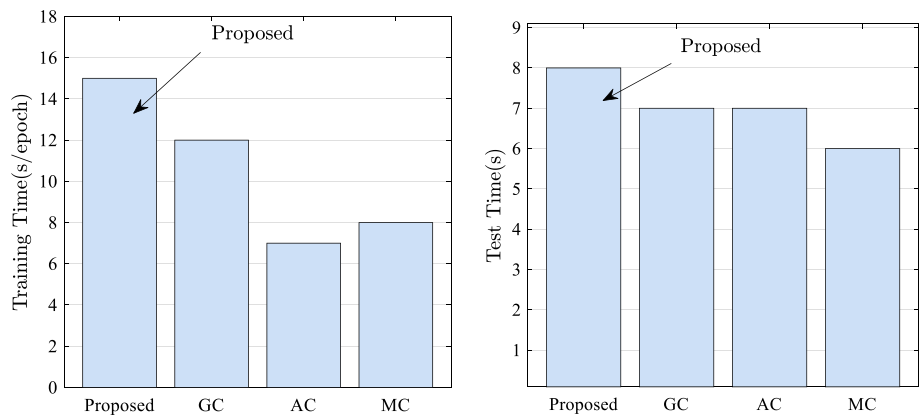


**Fig. 13** The classification accuracy under noise mismatch (Gaussian in training and non-Gaussian in test)



**Fig. 14** The box plot under noise mismatch

In Fig. 15, the proposed classification model is slightly inferior to other models in training and test speed, because the proposed has the maximum depth and more parameters. Since the difference is small, it can be considered comparable to other models.



**Fig. 15** Comparison with other methods in classification speed

**Table 3** Model size comparison

Algorithm type	Total parameters	Parameter size (MB)	Total size (MB)
Proposed	10947632	41.76	109.52
GC	10327880	39.32	100.23
AC	14597832	55.68	61.82
MC	4212280	16.06	117.42

## 5 Conclusions and future work

In this paper, an AMC method based on KD-GSENet is proposed to improve the performance of AMC, especially in the case of noise mismatch. Using the KD-tree to obtain more effective classification features, signals are preprocessed into color images. To capture different hidden information from the enhanced constellations of different signals, attention blocks are introduced to learn more distinguishing features. Numerical results show that the proposed method is more robust than traditional methods even in the case of noise mismatch.

Shortening classification time and reducing model size are worthy of further study. Future work will focus on more lightweight classification models.

### Abbreviations

AMC	Automatic modulation classification
DL	Deep learning
ML	Machine learning
CNN	Convolutional neural network
$E_b/N_0$	Ratio of bit energy to noise power spectral density;
SNR	Signal-to-noise ratio
I/Q	In-phase/quadrature
A/P	Amplitude/phase
ASK	Amplitude shift keying
PSK	Phase shift keying
QAM	Quadrature amplitude modulation
LSTM	Long short-term memory
KD-tree	$k$ -dimensional tree
BPSK	Binary phase shift keying
4ASK	4 Amplitude shift keying
QPSK	Quadrature phase shift keying
OQPSK	Offset QPSK
8PSK	Eight phase shift keying
16QAM	16-Ary quadrature amplitude modulation
32QAM	32-Ary quadrature amplitude modulation
64QAM	64-Ary quadrature amplitude modulation
GGN	Generalized Gaussian noise
PDF	Probability density function
SE	Squeeze-and-Excitation
BN	Batch normalization
ReLU	Rectified linear unit
Adam	Adaptive moment estimation
GC	GoogLeNet constellation
AC	AlexNet constellation
MC	MobileNetV3 constellation
KD-GSENet	KD-GoogLeNet and Squeeze-Excitation
GSENet	GoogLeNet and Squeeze-Excitation

### Acknowledgements

The authors acknowledged the anonymous reviewers and editors for their efforts in constructive and generous feedback.

### Author contributions

LG was in charge of the major theoretical analysis, experimental simulation and paper writing. RG proposed a framework of the whole algorithm. YC gave suggestions for the organization. LY made full contribution in the acquisition of funding. All authors read and approved the final manuscript.

### Funding

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61901408.

### Availability of data and materials

The datasets used during the current study are available from the corresponding author on reasonable request.

### Declarations

#### Competing interests

The authors declare that they have no competing interests.

Received: 9 December 2022 Accepted: 16 June 2023

Published online: 24 June 2023

**References**

1. H. Zhang, F. Zhou, Q. Wu, W. Wu, R.Q. Hu, A novel automatic modulation classification scheme based on multi-scale networks. *IEEE Trans. Cognit. Commun. Netw.* **8**(1), 97–110 (2022)
2. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(1), 4–24 (2021)
3. S.R. Kulkarni, G. Lugosi, S.S. Venkatesh, Learning pattern classification—a survey. *IEEE Trans. Inf. Theory* **44**(6), 2178–2206 (1998)
4. O.A. Dobre, A. Abdi, Y. Bar-Ness, W. Su, Survey of automatic modulation classification techniques: classical approaches and new trends. *IEEE Commun. Lett.* **1**(2), 137–156 (2007)
5. H. Qu, X. Xu, J. Zhao, F. Yan, W. Wang, A robust hyperbolic tangent-based energy detector with gaussian and non-gaussian noise environments in cognitive radio system. *IEEE Syst. J.* **14**(3), 3161–3172 (2020)
6. H. Zhang, L. Yuan, G. Wu, F. Zhou, Q. Wu, Automatic modulation classification using involution enabled residual networks. *IEEE Wirel. Commun. Lett.* **10**(11), 2417–2420 (2021)
7. Xu, J.L., Su, W., Zhou, M.: Likelihood-ratio approaches to automatic modulation classification. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **41**(4), 455–469 (2011)
8. S. Huang, Y. Yao, Z. Wei, Z. Feng, P. Zhang, Automatic modulation classification of overlapped sources using multiple cumulants. *IEEE Trans. Veh. Technol.* **66**(7), 6089–6101 (2017)
9. L. Han, F. Gao, Z. Li, O.A. Dobre, Low complexity automatic modulation classification based on order-statistics. *IEEE Trans. Wirel. Commun.* **16**(1), 400–411 (2017)
10. O.A. Dobre, A. Abdi, Y. Bar-Ness, W. Su, Survey of automatic modulation classification techniques: classical approaches and new trends. *IEEE Commun. Lett.* **1**(2), 137–156 (2007)
11. M. Abu-Romoh, A. Aboutaleb, Z. Rezki, Automatic modulation classification using moments and likelihood maximization. *IEEE Commun. Lett.* **22**(5), 938–941 (2018)
12. M.W. Aslam, Z. Zhu, A.K. Nandi, Automatic modulation classification using combination of genetic programming and KNN. *IEEE Trans. Wirel. Commun.* **11**(8), 2742–2750 (2012)
13. O'Shea, T.J., Corgan, J., Clancy, T.C.: Convolutional radio modulation recognition networks. In: *Engineering Applications of Neural Networks*, pp. 213–226 (2016)
14. F. Meng, P. Chen, L. Wu, X. Wang, Automatic modulation classification: a deep learning enabled approach. *IEEE Trans. Veh. Technol.* **67**(11), 10760–10772 (2018)
15. S. Chang, S. Huang, R. Zhang, Z. Feng, L. Liu, Multitask-learning-based deep neural network for automatic modulation classification. *IEEE Internet Things J.* **9**(3), 2192–2206 (2022)
16. Z. Zhang, H. Luo, C. Wang, C. Gan, Y. Xiang, Automatic modulation classification using CNN-LSTM based dual-stream structure. *IEEE Trans. Veh. Technol.* **69**(11), 13521–13531 (2020)
17. H. Zhang, M. Huang, J. Yang, W. Sun, A data preprocessing method for automatic modulation classification based on CNN. *IEEE Commun. Lett.* **25**(4), 1206–1210 (2020)
18. S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M.M. Sebdani, Y.-D. Yao, Modulation classification based on signal constellation diagrams and deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(3), 718–727 (2019)
19. S. Zhou, Z. Yin, Z. Wu, Y. Chen, N. Zhao, Z. Yang, A robust modulation classification method using convolutional neural networks. *EURASIP J. Adv. Signal Process.* **2019**(1), 1–15 (2019)
20. Y. Kumar, M. Sheoran, G. Jajoo, S.K. Yadav, Automatic modulation classification based on constellation density using deep learning. *IEEE Commun. Lett.* **24**(16), 1275–1278 (2020)
21. S. Huang, Y. Jiang, Y. Gao, Z. Feng, P. Zhang, Automatic modulation classification using contrastive fully convolutional network. *IEEE Wirel. Commun. Lett.* **8**(4), 1044–1047 (2019)
22. X. Yan, G. Zhang, J. Luo, H.C. Wu, Y. Wu, A novel automatic modulation classifier using graph-based constellation analysis for  $M$ -ary QAM. *IEEE Commun. Lett.* **23**(2), 298–301 (2019)
23. X. Yan, G. Liu, H.C. Wu, G. Feng, New automatic modulation classifier using cyclic-spectrum graphs with optimal training features. *IEEE Commun. Lett.* **22**(6), 1204–1207 (2018)
24. Y. Zeng, M. Zhang, F. Han, Y. Gong, J. Zhang, Spectrum analysis and convolutional neural network for automatic modulation recognition. *Wirel. Commun. Lett. IEEE* **8**(3), 929–932 (2019)
25. A.P. Hermawan, R.R. Ginanjar, D.S. Kim, J.M. Lee, CNN-based automatic modulation classification for beyond 5G communications. *IEEE Commun. Lett.* **24**(5), 1038–1041 (2020)
26. K. Bu, Y. He, X. Jing, J. Han, Adversarial transfer learning for deep learning based automatic modulation classification. *IEEE Signal Process. Lett.* **27**(99), 880–884 (2020)
27. Rajendran, S., Meert, W., Giustiniano, D., Lenders, V., Pollin, S.: Deep learning models for wireless signal classification with distributed low-cost spectrum sensors, in *IEEE Transactions on Cognitive Communications and Networking*, pp. 433–445 (2018)
28. T. Huynh-The, C.H. Hua, Q.V. Pham, D.S. Kim, MCNet: an efficient CNN architecture for robust automatic modulation classification. *IEEE Commun. Lett.* **24**(4), 811–815 (2020)
29. T. Huynh-The, Q.-V. Pham, T.-V. Nguyen, T.T. Nguyen, D.B. Costa, D.-S. Kim, RanNet: learning residual-attention structure in CNNs for automatic modulation classification. *IEEE Wirel. Commun. Lett.* **11**(6), 1243–1247 (2022)
30. S. Huang, R. Dai, J. Huang, Y. Yao, Y. Gao, F. Ning, Z. Feng, Automatic modulation classification using gated recurrent residual network. *IEEE Internet Things J.* **7**(8), 7795–7807 (2020)
31. T.J. O'Shea, T. Roy, T.C. Clancy, Over the air deep learning based radio signal classification. *IEEE J. Sel. Topics Signal Process.* **12**(1), 168–179 (2017)

32. H. Zhang, L. Yuan, G. Wu, F. Zhou, Q. Wu, Automatic modulation classification using involution enabled residual networks. *IEEE Trans. Wirel. Commun.* **10**(11), 2417–2420 (2021)
33. F. Zhang, C. Luo, J. Xu, Y. Luo, An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation. *IEEE Commun. Lett.* **25**(10), 3287–3290 (2021)
34. Q. Zhou, R. Zhang, F. Zhang, X. Jing, An automatic modulation classification network for IoT terminal spectrum monitoring under zero-sample situations. *EURASIP J. Wirel. Commun. Netw.* **2022**(1), 1–18 (2022)
35. J. Ma, T. Qiu, Automatic modulation classification using cyclic correntropy spectrum in impulsive noise. *IEEE Wirel. Commun. Lett.* **8**(2), 440–443 (2019)
36. S. Luan, Y. Gao, J. Zhou, Z. Zhang, Automatic modulation classification based on cauchy-score constellation and lightweight network under impulsive noise. *IEEE Wirel. Commun. Lett.* **10**(11), 2509–2513 (2021)
37. J. Miller, J. Thomas, Robust detectors for signals in non-gaussian noise. *IEEE Trans. Commun.* **25**(7), 686–690 (1977)
38. Banerjee, S., Agrawal, M.: Underwater acoustic noise with generalized gaussian statistics: effects on error performance. In: 2013 Ocean Electronics (SYMPOL), pp. 1–8 (2013)
39. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015)
40. J.L. Bentley, Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**(9), 509–517 (1975)
41. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)
42. J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(8), 2011–2023 (2020)
43. L. Chen, M. Zhou, W. Su, M. Wu, J. She, K. Hirota, Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction. *Inf. Sci. Int. J.* **428**(8), 49–61 (2018)
44. Kingma, D., Ba, J. Adam: A method for stochastic optimization. *Comput. Sci.* (2014)
45. J.G. Proakis, *Digital Communications*, 3rd edn. (Prentice Hall, Upper Saddle River, 1995)
46. B. Sklar, *Digital Communications, Fundamentals and Applications* (Prentice Hall, Upper Saddle River, 1998)
47. Z. Zhu, A.K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications* (Wiley, Hoboken, 2015)
48. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2018)
49. Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., Le, Q.: Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324 (2019)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)