IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2020, accepted July 8, 2020, date of publication July 16, 2020, date of current version July 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009471

A Hybrid Neural Network for Fast Automatic Modulation Classification

RENDENG LIN[1,2,3], WENJUAN REN[1,2], XIAN SUN[1,2,3], ZHANPENG YANG[1,2], AND KUN FU[1,2,3]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China
[2] Key Laboratory of Network Information System Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Wenjuan Ren (wjren2011@mail.ie.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61725105.

ABSTRACT Automatic modulation classification (AMC) plays a key role in cognitive radio. For AMC, convolutional neural networks (CNNs) have been explored in previous works extensively and deliver the best performance. However, temporal dependencies of signals modeled by CNNs are inherently implicit and insufficient. As a result, models need more data to learn discriminative features automatically. In this work, we propose a hybrid model named HybridNet, where a bidirectional gated recurrent unit (Bi-GRU) is placed after CNN to capture temporal dependencies explicitly. In addition, we investigate why varying Signal-to-Noise Ratio (SNR) dataset makes performance deteriorate. By visualization, we discover that the increase of the intra-class divergence under sharply varying SNR is the central cause. To this end, channel-wise attention is adopted in HybridNet to learn different patterns existing in SNR, which does not require SNR labels in the training process or inference values of SNR. On RadioML2016.10b, our HybridNet obtains the best accuracy among all scales of training data. Especially, in small datasets, our model obtains 87.4% accuracy that is 9.7% higher than the baseline method.

INDEX TERMS Cognitive radio, modulation recognition, deep learning, signal-to-noise ratio.

I. INTRODUCTION

Automatic modulation classification (AMC) is a key component in intelligent wireless communication, which is widely used in cognitive radio and military electronic warfare. In dynamic spectrum access, modulation classification is applied to identify radio sources and avoid interference in the vicinity to enhance spectral efficiency [1]. In warfare, it helps to identify hostile or friendly signals without prior information.

There are two categories of traditional modulation classification methods, likelihood-based approaches and feature-based approaches. In likelihood-based approaches, the likelihood ratio is computed by the probability density function of the observed wave and compared with a threshold [2], [3]. In feature-based approaches, several features are extracted by hand and hard boundaries or classic pattern recognition methods are utilized to make a decision [4], [5]. However, with the explosion in data scale and the evolution of modulation technology, simple methods have exposed several problems. For example, the accuracy decreases in multi-class tasks and the data is not efficiently utilized.

Recently, convolutional neural networks (CNNs) have made a series of breakthroughs in computer vision [6]–[8]. The success of CNN based neural networks raises more attention in AMC. O'Shea *et al.* [9] firstly proposed a CNN for classifying modulation types, which showed the CNN based method was superior to traditional methods and more robust for a variety of signal types. Ramjee *et al.* [10] explored several mainstream neural structures, among which ResNetSig obtained the highest accuracy with adequate data. Wang *et al.* [11] proposed a model integrating two CNNs, where the former was trained on samples composed of in-phase and quadrature components to classify easy types while the posterior one learned from constellation diagrams to identify QAM16 and QAM64.

Previous works have made significant contributions to the exploration of CNN structure in AMC and obtain pretty

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

good results. However, temporal dependencies of signals modeled by CNNs are inherently implicit and insufficient. CNNs fuse both spatial and channel-wise information within local receptive fields at each layer to construct discriminative features. To capture long term dependencies, CNNs need wider receptive fields. Stacking convolutional layers is the way for CNNs to broaden receptive fields. Meanwhile, it pushes the net to deeper and more data is needed for training. Recurrent neural network (RNN) is simple and effective in modeling long term dependencies. In [10], [12], a two-layer Long Short Term Memory (LSTM) obtained comparable accuracy to ResNetSig at high SNR and outperformed CNNs significantly with small data. Although RNNs can capture temporal information efficiently, their time costs are unacceptable and models are not as robust as the CNNs at varying SNR. The mixed structure of RNN and CNN was widely used in the image caption, video classification and speech recognition, in which CNN was exploited as a feature extractor and RNN was responsible for modeling the temporal dependencies [13]–[15]. Convolutional Long Short Term Deep Neural Network (CLDNN)[15] is a typical kind of mixed structure that integrated CNN layers and LSTM layers for speech recognition. Then the same structure [10] was introduced into AMC, but it didn't show any advantages in speed or accuracy. We argue that the capacity of the mixed structure is limited by unsuitable configurations, such as the lack of pooling operations, too many channels on CNN. We propose an efficient hybrid structure, named HybridNet, that outperforms the state of the art accuracy.

In addition, we investigate the reason why the training set with various SNR makes performance deteriorate. The strength of noise in the signal is described as Signal-to-Noise Ratio (SNR). In [16], authors noticed negative impacts of samples with low SNR during training and proposed an SNR-aware loss, by which high-quality samples denoted as high SNR were emphasized and weights of samples with low SNR were suppressed. Although this strategy improved the accuracy at high SNR, it brought the decrease at low SNR and required SNR labels. Xie *et al.* [17] proposed a multi-branch model for varying SNR environments, in which an M2M4 algorithm was utilized to estimate SNR and then a specific branch was selected for AMC according to the value of SNR. That model isolated the interference of SNR through different branches. But it enlarged the number of parameters and required more samples for training because different SNR samples were used in a non-shared manner. Why samples with sharply different SNR affect the training process remains unknown.

In this paper, to improve the ability of CNN in capturing temporal information of the signal, we propose the HybridNet, where the bidirectional gated recurrent unit [18] (Bi-GRU) is used after CNN to model the temporal dependencies explicitly. To understand the effects of noise, we visualize distributions of samples at each SNR using principal component analysis (PCA). We discover that the distribution shift caused by noise results in the increase of the intra-class divergence, which puts a severe challenge on training. In this case, more samples are required to match the divergence. Squeeze-and-Excitation (SE) [19] block is an implementation of channel-wise attention, which enables the net to model differences among samples by recalibrating channel responses. We adopt channel-wise attention in HybridNet to learn different patterns existing in SNR without adding data or SNR labels. Our main contributions are summarized as the following three points.

Firstly, we propose the HybridNet to make up the insufficiency of CNN in capturing temporal information of the signal, in which the Bi-GRU is used after CNN to capture temporal dependencies of the signal explicitly and a dual-classifier is applied to produce predictions.

Secondly, we introduce channel-wise attention to model different patterns existing in SNR for mitigating interference brought by noise.

Thirdly, a thorough evaluation is done on RadioML2016. 10b. Our HybridNet achieves the best performance at various data scales. Especially, our model is 9.7% higher than the baseline in small datasets.

The rest of this paper is organized as follows. Section II details problems existing in current CNNs. Section III presents our HybridNet and intuitive understandings behind it. Experimental settings and results are shown in Section IV. Ablation experiments in Section V reveal the effects of each part in HybridNet. In Section VI, we summarize the whole work.

II. PROBLEM STATEMENT

Two main problems are existing in current CNNs for AMC. One is the absence of capturing temporal dependencies of the signal. Another is the interference of noise in the training process. Without considering those problems, CNNs need more data to match intricate distributions by themselves. We describe those problems in detail as follows.

A. MODELING TEMPORAL DEPENDENCIES

In general, CNNs exploit CNN as a feature extractor and place fully connected layers subsequently to produce the predictions. We consider both of them are incompetent to model the temporal dependencies.

The essence of CNNs is the convolution operator, which fuses both spatial and channel-wise information within local receptive fields to produce informative features. The convolution operator maps an input $X \in R^{N\times H\times W}$ with N channels to features $U \in R^{M\times H\times W}$. We use $V = [v_1, v_2, \ldots v_M]$, $V \in R^{M\times N\times k\times k}$, to denote the convolutional kernels and formulate the transformation as (1).

$$u^m = v_m * X = \sum_{i=1}^{N} v_m^i * x^i \tag{1}$$

where $*$ denotes the convolution, $X = [x^1, x^2, \ldots, x^N]$, $U = [u^1, u^2, \ldots, u^M]$. $v_m = [v_m^1, v_m^2, \ldots, v_m^N]$, $v_m^i \in R^{k\times k}$, k is the size of convolutional kernels. Receptive fields of units in U are limited by both the kernel size k and input X. Deeper structure and bigger kernel size are helpful to broaden the receptive fields. Benefiting from the local correlation captured by kernels and the hierarchical structure of networks, temporal dependencies are implicitly embedded into feature maps. However, temporal dependencies captured by CNNs are inherently inefficient because this manner is hard to capture long term dependencies (e.g. two ends of the sample) and requires deeper structure.

The fully connected layer is able to model long term dependencies because of global receptive fields, but it is dragged down by numerous parameters. It operates all units of input simultaneously so that it is hard to concentrate on useful parts without sufficient data.

RNN is direct and effective in modeling temporal dependencies. We propose a HybridNet, where a Bi-GRU is placed after CNN subsequently to grasp the temporal dependencies without stacking deeper layers.

B. THE DIVERGENCE LED IN BY NOISE

Noise increases the intra-class divergence and narrows the gap of inter-class, both of which make a severe challenge in classification. For electronic signals, a variety of noise exists inherently and is inevitable. To quantitative analysis, SNR is adopted as the measure of the quality of signal in communication systems, shown as (2).

$$SNR = 10 \times lg\frac{P_S}{P_N} \tag{2}$$

P_S and P_N represent the effective power of signal and noise, respectively. Typically, it's more difficult to extract information from low SNR samples than high SNR ones. Intuitively, we made a hypothesis that the distribution suffers different degrees of shift according to SNR, which increases the intra-class divergence. Taking the extreme situation as an example, at low SNR (e.g. -20dB), information of modulation has been submerged in noise, which means all categories with -20dB only show the distribution of noise and can't be classified correctly. To examine the assumption, we visualize GFSK and WBFM with different SNR using principal component analysis (PCA). Specifically, we extract two hundred of samples from each class at each SNR randomly and concatenate in-phase and quadrature components as the input of PCA. Shown as Fig. 1, although samples hold the same modulation type, there are apparent differences in distributions of each SNR. Meanwhile, all the categories form a cluster at the center when SNR is equal to -8 dB. Those observations confirm our hypothesis.

Within the same category, different SNR samples hold diverse distributions, imposing interference with each other. We note that training ResNetSig on the full range of SNR (SNR $\in [-20, 20]$), the curve of training loss oscillates heavily. Once low SNR (SNR > 0 dB) samples are removed from the training set, the instability is mitigated. Another observation is that accuracy of the model trained on high SNR is about 1% higher than the ones trained at the full range of SNR. Those observations suggest that modeling the divergence led in by noise would improve the performance of models.

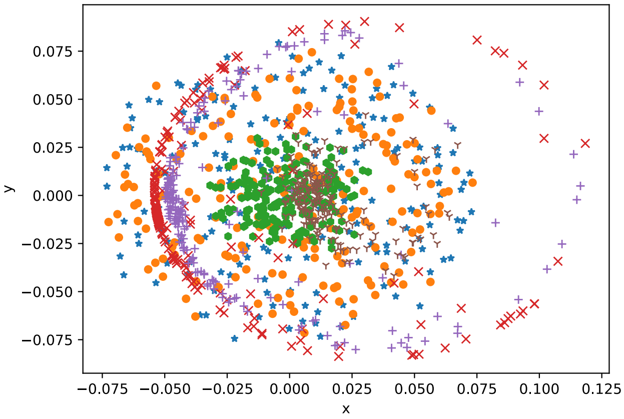


FIGURE 1. Visualization of WBFM and GFSK using PCA. Naming scheme: Modulation_SNR(dB).

To this end, channel-wise attention that is able to model the divergence existing among samples is utilized in HybridNet to learn different patterns varied by SNR.

III. ARCHITECTURE

As shown in Fig. 2, our HybridNet comprises a feature extractor, a bidirectional gated recurrent unit (Bi-GRU) and a dual-classifier. For each input with the form of IQ components, it produces the probabilities of categories.

A. CNN BASED FEATURE EXTRACTOR

For the robustness of the model over the full range of SNR, HybridNet applies a convolutional neural network as the feature extractor. The feature extractor is composed of a stack of 3 basic blocks. Each block consists of a pre-filter, a standard residual block [8], two Squeeze-and-Excitation (SE) blocks and a maxpooling layer, see Fig. 2. All of the convolutional layers use the same filter size of 1×3.

The pre-filters have two effects in HybridNet. Firstly, they are used to convert the input into specific dimensions so that the residual blocks can use identity mapping simply without dimensional projection. In addition, the pre-filters avoid introducing the raw distribution, which is similar to the first 7×7 filter adopted in ResNet [8]. ResNetSig [10] used 1×1 filters for matching dimensions while it introduced raw distributions into subsequent layers. That is the reason why training loss of ResNetSig was unstable. ReLu [19] activation function and batch normalization [24] are used for all CNN layers. Maxpooling operations provide the shift-invariant features that are required for fully connected layers. Meanwhile, it reduces the feature size so that the cost of time in GRU is mitigated.

The squeeze and excitation (SE) block is placed after each pre-filter and residual block to model the intra-class divergence. SE block [19] is a kind of channel-wise attention that investigates interdependencies between channels.

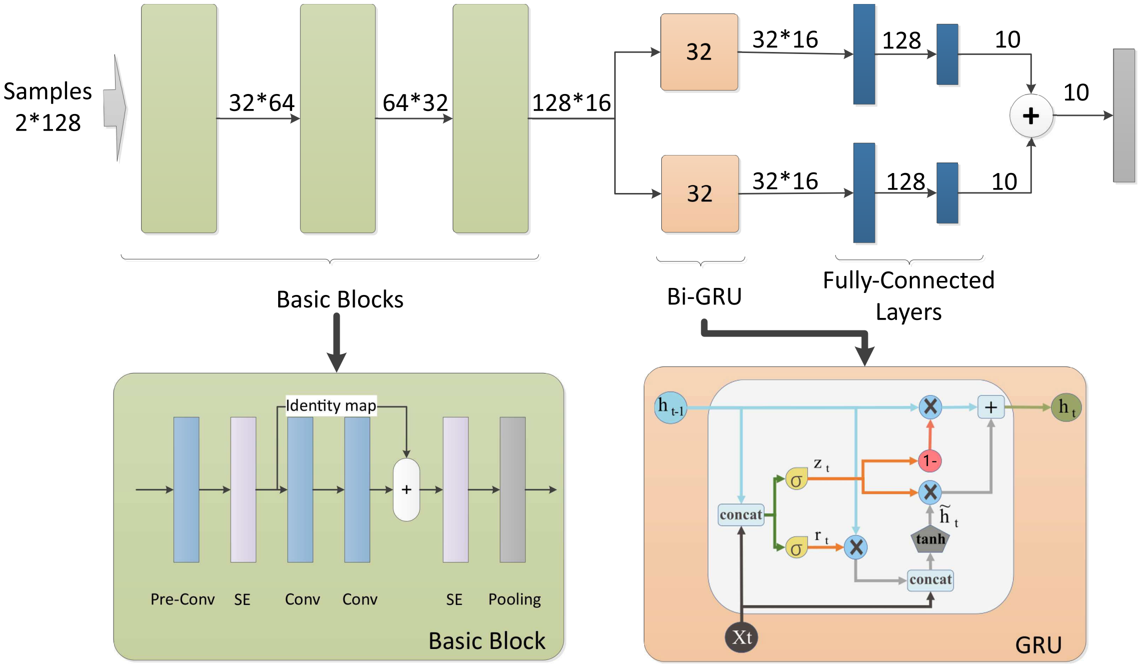


FIGURE 2. **HybridNet-b. HybridNet comprises a feature extractor, a bidirectional gated recurrent unit (Bi-GRU) and a dual-classifier. Features extracted by CNN are fed to Bi-GRU for temporal dependencies. Then dual branches of classifier operate forward and backward output of Bi-GRU individually to produce the prediction.**

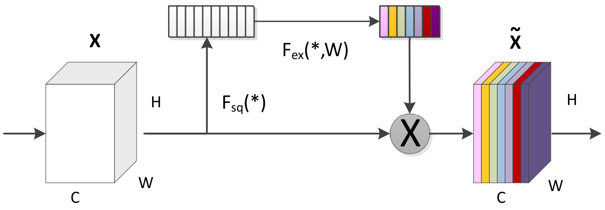


FIGURE 3. **Squeeze and Excitation Block (SE).**

The central of SE is squeeze and excitation operation, see Fig. 3. The squeeze operation aggregates global information of each channel in the input $X \in R^{N\times H\times W}$ to form channel descriptors $C = [c_1, c_2, \ldots c_N]$, formulated as (3). The excitation operation consists of two fully connected layers, formulated as (4). It maps channel descriptors into a set of weights to emphasize more informative channels.

$$c_i = F_{sq}(x_i) = \frac{1}{HW}\sum_{m=1}^{H}\sum_{n=1}^{W} x_i(m,n) \tag{3}$$

$$w_i = F_{ex}(C, W) = \sigma(W_2 ReLu(W_1 C)) \tag{4}$$

$$\tilde{x_i} = w_i * x_i \tag{5}$$

Here $C \in R^{N\times 1\times 1}$, $W = [W_1, W_2]$ are learnable parameters of fully connected layers. σ refers to the logistic sigmoid function and ReLu [23] is a activation function. $X = [x_1, x_2, \ldots, x_N]$ are input feature maps. SE blocks re-weight the input X to produce the output $\widetilde{X} = [\tilde{x_1}, \tilde{x_2}, \ldots, \tilde{x_k}]$. With the increasing depth, the manner that SE blocks activate feature channels changes from a class-agnostic to a highly class-specific. In our model, it excites different feature channels according to input, which enables samples with varying values of SNR to rely on diverse channels to avoid interference.

B. THE BI-GRU AND CLASSIFIER

To make up the insufficiency of CNN in capturing temporal information of the signal, we adopt a bidirectional gated recurrent unit to model the temporal dependencies explicitly. Gated recurrent unit (GRU) [14] is a typical kind of RNN, which operates on a variable-length sequence, $x = [x_1, x_2, \ldots, x_T]$, to learn the probability over the sequence. At each time step t, it receives the x_t and previous state h_{t-1} as inputs to produce the hidden state h_t.

$$h_t = f(h_{t-1}, x_t) \tag{6}$$

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \tag{7}$$

$$z_t = \sigma(W_z[h_{t-1}, x_t]) \tag{8}$$

$$\widetilde{h_t} = tanh(W_{\tilde{h}}[r_t \odot h_{t-1}, x_t]) \tag{9}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t} \tag{10}$$

f is a non-linear activation function that is implemented by a reset gate r_t and an update gate z_t, shown in Fig. 2. W_r, W_z, $W_{\tilde{h}}$ are parameters in corresponding gates and σ denotes sigmoid function. $\odot$ denotes Hadamard product. At each time t, r_t and z_t are computed by previous state h_{t-1} and input x_t. The reset gate and update gate are sensitive to shorter and longer term dependencies respectively so that GRU is able to capture dependencies over different scales.

In our HybridNet, the features produced by the extractor are sent into a bidirectional GRU to capture the temporal dependencies. At each time, the Bi-GRU produces two components, the forward component is calculated from front to end and the backward component is the opposite. Each input to the Bi-GRU has the same receptive fields and may be equally important. After the GRU, post-computed components access to larger fields and therefore are more informative. The forward and backward components pass through an unshared classifier that is two fully connected layers to generate predictions respectively. Then, two results are added together and category probabilities are calculated with softmax.

C. MODEL PARAMETERS

We have designed two models, HybridNet-big and HybridNet-small, both of which share the same structure but are different in the parameter size. In the convolutional part, HybridNet-b follows the idea of VGG [6], which doubles the number of filters once the feature map size is halved. HybridNet-s is conformant with the ResNetSig [10] using fixed 32 channels. HybridNet-b has 32 hidden units in each of GRUs, while HybridNet-s reduces the number of hidden units to half, 16.

IV. EXPERIMENTS

In this section, we introduce the dataset used for evaluation. Then, implementation details are described, including the re-implementation of three baseline models reported in [10]. Results obtained over different scales of training datasets are presented.

A. DATASET

We use the RadioML2016.10b [20] generated with GNU Radio as our dataset in our experiments. The dataset has 1.2M samples that consist of eight digital modulations and two analog modulations. Eight digital modulations contain BPSK, QPSK, 8PSK, QAM16, QAM64, BFSK, CPFSK and PAM4. Two analog modulations are WBFM and AM-DSB. A variety of noise is considered in the dataset, including moderate LO drift, light fading and Gaussian white noise in different intensity. Each sample has a length of 128 and is stored as a complex array, in which the real and imaginary parts represent I and Q component respectively. The label consists of two elements, the modulation type and the SNR value. SNR ranges from −20dB to 20dB with the step of 2dB.

B. IMPLEMENTATION DETAILS

1) DATA AUGMENTATION

We use random flipping and cropping in our experiments, which are commonly used in computer vision [6] and efficient to enhance generalization. Flipping operation reverses the sample from beginning to end. Random cropping selects continuous K-length from the N-length sample as the input. Usually, to keep the length of the input, interpolation is always applied to expand dimensions before cropping. Different from interpolation methods used in CV, we argue that FFT interpolation [21] is a better choice for the signal. FFT interpolation that inserts zeros between the negative and the positive frequency components is a non-destructive operation and mitigates the fence effect. Other methods are just fitting the distribution. FFT interpolation can not create extra information and cropping discards a part of the information, according to the proportion of cropping. So expanding too much is not recommended.

2) TRAINING SETTINGS

To evaluate the performance of each model, we split the whole dataset into training and test set equally. To assess the efficiency of models, we conduct experiments on different sizes of the training sets (20000, 100000 and 600000) and calculate test accuracies at samples in the test set whose value of SNR are higher than 0dB. We adopt Adam [22] solver with a batch size of 1024 in most cases and decrease the batch size to 256 under 20,000 samples. The learning rate starts with 0.01 and is decayed by 0.1 once the number of epoch reaches one of the milestones, [20, 50, 70]. The number of epochs is set to 100. Random flipping and cropping are default preprocessing operations in all experiments. For cropping, each sample is expanded to 140 points using FFT interpolation and then cropped into 128 points randomly. The default probability of flipping is 0.2. All experiments are implemented in Pytorch and conducted on an Nvidia K80s GPU to speed up training. For stable results and statistical significance, all experiments using 20,000 training samples are repeated ten times. And then the average is taken as the final value. For experiments using 0.6M training samples, results are averaged by three repeated experiments.

C. BASELINE MODELS

To enable a fair comparison, We re-implement three models reported in [10], namely LSTM, CLDNN and ResNetSig. Our re-implementations use the same configuration with our model and obtain better accuracies than results reported in the original paper. Originally, LSTM took the best accuracy under a small dataset followed by CLDNN and ResNetSig. In our case, CLDNN gets the best and ResNetSig is better than LSTM. We think that the rank changes because training settings are more suitable for CNN-based models than LSTM, such as random cropping.

D. RESULTS

We evaluate each model trained at different data scales to assess efficiency. As shown in Table 1, test accuracies of HybridNet outperform previous works in all tests. With the decrease of data size, the improvement of accuracy is becoming obvious. Especially, with 20,000 training samples, the accuracy of HybridNet-b is 9.7% higher than ResNetSig. Under 20,000 training samples, we visualize the confusion matrix at 10dB that is obtained by HybridNet-b using the test set. As shown in Fig. 5, accuracies of most categories

TABLE 1. Test accuracy (%) and Speed. N is the number of training samples. Accuracies of models are calculated by samples in the test set whose value of SNR is higher than 0dB. The speed refers to the time consumption of one epoch when the model is trained in 0.6M samples.

samples size models	N=0.6M	N=0.1M	N=20K	speed(s)	parameters(M)
ResNetSig[10]	92.7	89.7	77.7	54	0.148
LSTM[10]	92.0	82.0	74.9	352	1.167
CLDNN[10]	91.6	83.1	80.3	244	1.323
HybridNet-b	93.6	92.6	87.4	87	0.345
HybridNet-s	93.4	91.7	85.2	50	0.098
HybridNet-b No Aug	93.5	90.7	84.3	84	0.345

TABLE 2. Statistical Significance (%).

models	ResNetSig[10]	LSTM[10]	CLDNN[10]	Our-b	Our-s	Our-b No Aug
mean	77.7	74.9	80.3	87.4	85.2	84.3
std	4.67	0.83	2.04	1.05	1.08	0.78

are higher than 97% and misclassifications are mainly concentrated in two pairs of categories, QAM16 and QAM64, WBFM and AM-DSB. Those confusions also happen under enough training data but are more serious in small datasets. In the convolutional part, HybridNet-s have the same number of channels as ResNetSig and fewer layers. Accuracies of HybridNet-s are significantly better than baseline models among all tests of different scales. It suggests that our improvements are based on utilizing temporal information and attention mechanism rather than using more feature maps.

For statistical significance, we conduct ten repeated experiments for models trained in 20,000 samples and list means and standard deviations of test accuracy in Tab 2. Results suggest that our model is significantly better than baselines and has better stability in varying SNR dataset. Because leading the raw distribution of samples into deeper layers, ResNetSig is affected by noise severely and its standard deviation behaves as an outlier.

Evaluated in 600,000 training samples, HybridNet-s is the fastest model in our experiments whose training time is 50s per epoch despite time-consuming RNN. Although HybridNet-b is better than HybridNet-s marginally at accuracy, it costs much more time than the small one. Weighing the cost of time and accuracy, we consider HybridNet-s is comparable to HybridNet-b and doubling the number of channels after pooling is not an essential operation.

Training on 20,000 samples, over-fitting can be observed in all models. However, we do not take any measures to deal with it but data augmentation. The last line in Tab 1 is the results of HybridNet-b without data augmentation methods. Those methods contribute to enhancing the generalization mainly and improve test accuracy. Without it, training accuracies reach an unrealistic point and over-fitting is serious. The flipping operation in data augmentation plays a role in preventing overfitting. Indeed, it mixes the forward and backward temporal dynamics, which makes the net do not overfit to the training set. We conduct experiments to show the performance of our model in different values of the probability of flipping. Results have been listed in Tab 3 and curves of training and validation accuracy are plot in Fig. 4. Because of temporal confusion led in by flipping operation, evenly flipping is showed as underfitting that will degrade the accuracy. As the probability goes from 0 to 0.5, the model will transition from overfitting to underfitting. In our experiments, we set the probability to 0.2, a good balance between underfitting and overfitting.

TABLE 3. Flipping with different probabilities. The model,HybridNet-b, is trained in 20,000 samples and evaluated by samples whose SNR are greater than 0dB.

probability	0.0	0.2	0.5
mean(%)	86.5	87.4	86.4
std	1.12	1.05	1.84

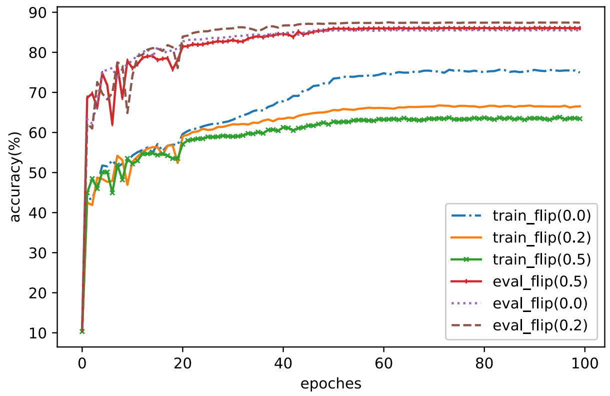


FIGURE 4. Training accuracy is calculated by 20,000 samples whose SNR range from −20dB to 20dB. Validation accuracy is obtained by 270,000 samples whose SNR are greater than 0dB. Note that train-accuracy and eval-accuracy are calculated using different ranges of SNR. So, there is a significant gap.

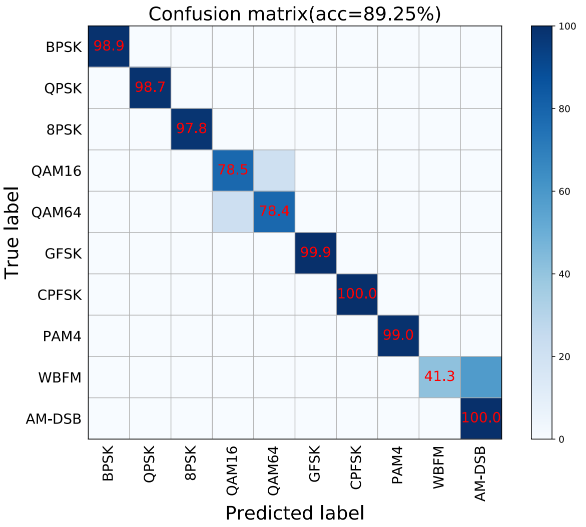


FIGURE 5. With SNR value of 10 dB, confusion matrix is obtained by HybridNet-b using the test set. The model is trained by 20,000 samples whose SNR values range from −20dB to 20dB.

V. ABLATION

In this section, we conduct ablation experiments to investigate the effects of each part on HybridNet-b. The training configuration follows the description in Section IV-B.

A. SE BLOCK

Improvements brought by SE blocks are evaluated on different data scales, shown in Tab 4. With 600,000 training samples, improvement of accuracy is about 0.8%, and increases to 2% under 20,000 samples. We consider that sufficient data enables the network to eliminate interference automatically, while explicit modeling is more effective under a limited data scale.

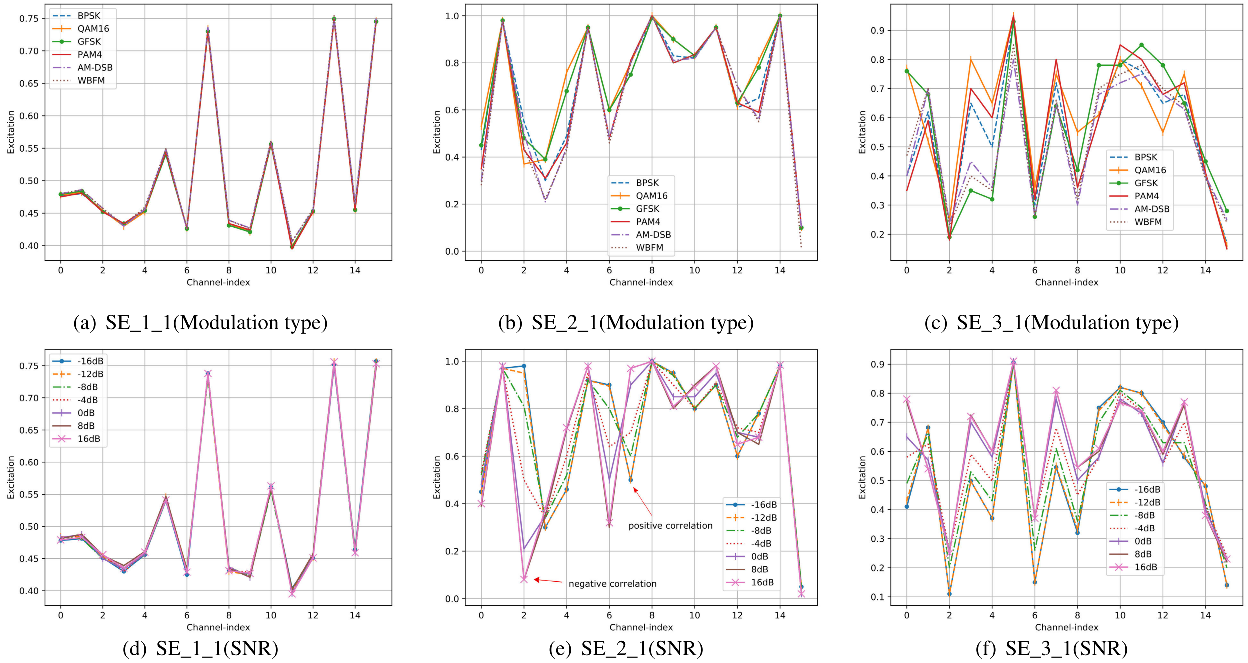
TABLE 4. Ablation of SE blocks.

models	0.6M	0.1M	20K
SE	93.6	92.6	87.4
No SE	92.8	90.4	85.4

To reveal how SE block improves the performance, we plot distributions with respect to different classes and different SNR of each basic block in HybridNet, shown in Fig. 6. Specifically, we extract 500 samples randomly for each SNR from the test set and compute the average of activations for the first sixteen channels. The distributions of activations of classes are obtained in the same manner.

As shown in Fig. 6, SE block not only is sensitive for different modulation types (see Fig. 6(a-c)) but also responds to varying SNR (see Fig. 6(d-f)). We make two observations from visualization. First, distributions across different modulation types or SNR are very similar in the first block, which is consistent with the conclusion drew in [19] that the importance of feature channels is likely to be shared by different classes in the early stages. The second is that distributions across different SNR samples or modulation types exhibit different preferences to feature maps at deeper layers.

For response for different modulation types, channels show category-related characteristics. Especially, channels' responses to analog modulation types, AM-DSB and WBFM, are highly coincident, shown as Fig. 6(b).



(a) SE_1_1(Modulation type) (b) SE_2_1(Modulation type) (c) SE_3_1(Modulation type)

(d) SE_1_1(SNR) (e) SE_2_1(SNR) (f) SE_3_1(SNR)

FIGURE 6. Channel responses. Naming scheme: SE_BlockID_LayerID. Subfigures (a-c) show distributions of excitation for different modulation types. Subfigures (d-f) show how SE blocks respond to varying SNR. With increasing depth, the excitations become increasingly class-specific or SNR-specific.

TABLE 5. The effect of temporal information.

models	0.6M	0.1M	20K
Bi-GRU	93.6	92.6	87.4
GRU	93.6	92.4	86.2
No GRU	93.5	92.3	84.6

For response for varying SNR samples, some channels' responses are highly correlated with SNR, while the others are insensitive to SNR. For example, at Fig. 6(e), the strength shows a negative correlation with SNR at 2th channel and exhibits a positive correlation at the 7th channel. When the value of SNR is greater than 0dB, their distribution curves almost coincide. Channels that are insensitive to SNR are likely to work in a shared way, while the others affiliate to different SNR individually. These observations suggest that the network assigns different patterns for sharply different SNR. So that samples with sharply different SNR propagate the gradient along different paths, which is a little similar to a multi-branch net. That may be why channel-wise attention is able to eliminate the interference and improve accuracy.

B. BI-GRU

To assess the importance of temporal information, we compare the differences between the same model adding Bi-GRU or without it. Results shown in Tab 5 suggest that using GRU to grasp the temporal information is helpful to improve the accuracy of models, especially under small datasets. With enough data, temporal dependencies can be embedded in feature maps or captured by fully-connected layers implicitly. Under small datasets, CNNs can also achieve high training accuracy, but they are hard to generalize in the test sets. At this time, the model with GRU is significantly better than CNNs, which shows that temporal dependencies are essential properties.

We use bidirectional GRU to improve generalization. As shown at line 2 in the Tab 5, compared to Bi-GRU, a single GRU with 64 hidden units achieves similar results and is sufficient for most cases. Bi-GRU that provides different features obtained from forward and backward direction contributes to generalization in small datasets.

VI. CONCLUSION

In this work, we propose the HybridNet, in which a Bi-GRU is placed after the CNN based feature extractor to capture the long term dependencies of the signal. In addition, we figure out that noise makes the distribution shift and increases the intra-class divergence. Then we introduce channel-wise attention to model the divergence. A thorough evaluation on RadioML2016.10b shows the effectiveness of our model, which obtains the best result over diverse data scales.

REFERENCES

[1] M. Höyhtyä, A. Mammela, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2386–2414, 4th Quart., 2016.

[2] J. A. Sills, "Maximum-likelihood modulation classification for PSK/QAM," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Atlantic City, NJ, USA, vol. 1, Oct./Nov. 1999, pp. 217–220.

[3] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1199–1211, Aug. 1990.

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

[5] A. Ali and F. Yangyu, "Higher-order statistics based modulation classification using hierarchical approach," in *Proc. IEEE Adv. Inf. Manage., Commun., Electron. Automat. Control Conf. (IMCEC)*, Xi'an, China, Oct. 2016, Art. no. 370374.

[6] K. Simonyan and A. Zisserman, "Deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[9] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Cham, Switzerland: Springer, 2016, pp. 213–226.

[10] S. Ramjee, S. Ju, D. Yang, X. Liu, A. El Gamal, and Y. C. Eldar, "Fast deep learning for automatic modulation classification," 2019, *arXiv:1901.05850*. [Online]. Available: http://arxiv.org/abs/1901.05850

[11] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[12] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.

[13] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, Apr. 2017.

[14] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 461–470.

[15] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4580–4584.

[16] J. Nie, Y. Zhang, Z. He, S. Chen, S. Gong, and W. Zhang, "Deep hierarchical network for automatic modulation classification," *IEEE Access*, vol. 7, pp. 94604–94613, 2019.

[17] X. Xie, Y. Ni, S. Peng, and Y.-D. Yao, "Deep learning based automatic modulation classification for varying SNR environment," in *Proc. 28th Wireless Opt. Commun. Conf. (WOCC)*, Beijing, China, May 2019, pp. 1–5.

[18] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[19] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[20] T. O' Shea and N. West, "Radio machine learning dataset generation with GNU radio," in *Proc. GNU Radio Conf.*, 2016, pp. 1–6.

[21] K. P. Prasad and P. Satyanarayana, "Fast interpolation algorithm using FFT," *Electron. Lett.*, vol. 22, no. 4, pp. 185–187, Feb. 1986.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[23] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 1–9.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

RENDENG LIN received the B.Sc. degree from Chongqing University, Chongqing, China, in 2018. He is currently pursuing the M.S. degree with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. His research interests include signal processing, pattern recognition, and machine learning.

WENJUAN REN received the B.Sc. degree from Xidian University, Xi'an, China, in 2005, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2011. She is currently an Associate Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences. Her research interests include signal processing, radar technology, communication engineering, and data mining.

XIAN SUN received the B.Sc. degree from Beihang University, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2006 and 2009, respectively. He is currently a Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests include deep learning, computer vision, and remote sensing image understanding.

ZHANPENG YANG received the Ph.D. degree from the Beijing Institute of Technology, in 2016. From 2016 to 2020, he was a Postdoctoral Scholar with the Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests include functional analysis, signal processing, and differential equations.

KUN FU received the B.Sc., M.Sc., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1995, 1999, and 2002, respectively. He is currently a Professor with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China. His research interests include deep learning, remote sensing image understanding, geospatial data mining, and visualization.