Received: 23 July 2020 | Revised: 28 August 2020 | Accepted: 19 September 2020

DOI: 10.1002/int.22302

RESEARCH ARTICLE

WILEY

One-dimensional convolutional neural networks for high-resolution range profile recognition via adaptively feature recalibrating and automatically channel pruning

Qian Xiang[1] | **Xiaodan Wang**[1] | **Yafei Song**[1] | **Lei Lei**[2] | **Rui Li**[1] | **Jie Lai**[1]

[1]College of Air and Missile Defense, Air Force Engineering University, Xi'an, China

[2]College of Information and Navigation, Air Force Engineering University, Xi'an, China

Correspondence
Xiaodan Wang, College of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China
Email: afeu_wang@163.com

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 61806219, 61876189, 61503407, 61703426, 61273275; Young Talent Fund of University Association for Science and Technology in Shaanxi, China, Grant/Award Number: 20190108; Innovation Talent Supporting Project of Shaanxi, China, Grant/Award Number: 2020KJXX-065

Abstract

High-resolution range profile (HRRP) has obtained intensive attention in radar target recognition and convolutional neural networks (CNNs) are among predominant approaches to deal with HRRP recognition problems. However, most CNNs are designed by the rule-of-thumb and suffer from much more computational complexity. Aiming at enhancing the channels of one-dimensional CNN (1D-CNN) for extracting efficient structural information oftargets form HRRP and reducing the computation complexity, we propose a novel framework for HRRP-based target recognition based on 1D-CNN with channel attention and channel pruning. By introducing an aggregation-perception-recalibration (APR) block for channel attention to the 1D-CNN backbone, channels in each 1D convolutional layer can adaptively learn to recalibrate the extracted features for enhancing the structural information captured from HRRP. To avoid rule-of-thumb design and reduce the computation complexity

of 1D-CNN, we proposed a new method incorporated withthe global best leading artificial bee colony (GBL-ABC) to prune the original network based on the lottery ticket hypothesis in an automatic and heuristic manner. The extensive experimental results on the measured data illustrate that the proposed algorithm achievesthe superiorrecognition rate by combing APR and GBL-ABC simultaneously.

KEYWORDS
channel attention, channel pruning, convolution neural networks, global best leading artificial bee colony, high-resolution range profile

1 | INTRODUCTION

High-resolution range profile (HRRP) is the superposition of wideband radar echoes of target scattering centers on the corresponding range cell, representing a one-dimensional projection of the scattering intensity distribution onto the radar line-of-sight (RLOS). It not only provides the geometric and structural characteristics of the target but also contains abundant discriminative information for target recognition. Compared with two-dimensional synthetic aperture radarand inverse synthetic aperture radar images, which have obtained intensive attention in radar target recognition as well, HRRP has the advantages of easy acquisition and storing as well as low computation complexity. Therefore, it is widely used in recognition of a variety of radar targets, such as ballistic missiles,[1-4] ships,[5-7] tanks,[8,9] airplanes,[10-14] and so on, it has become a hotspot in the community of radar automatic target recognition (RATR).

In recent years, deep learning is among prevalent techniques to handle with target recognition tasks and extensively utilized for HRRP recognition due to their powerful feature expressive ability. Du et al.[15] utilize variational auto-encoder to learn discriminative latent features of HRRP by incorporating the label information and take the multilayer perception (MLP) as a sufficient statisticto improve recognition performance. Pan et al.[16] take advantage of t-distributed stochastic neighbor embedding (t-SNE) and synthetic sampling to acquire balanced HRRP data set, and then propose a discriminant deep belief network for recognition. Considering the dependence across each range cell, Liu et al.[17] introduce an attention mechanism to a bidirectional self-recurrent neural network to take advantage of temporal information of HRRP, thus achieving accurate identification in the circumstance of time-shifting.

Although aforementioned models based on fully connected (FC) networks have made remarkable progress in extracting effective and robust features, they have weakness in capturing the structural information among the range cells layer by layer, since HRRP reflects the distribution of scatters in target along the range dimension. In FC networks, feature maps collected in the posterior layers are just the weighted and biased summation of that in the anterior layers. The connectivity of convolutional neural networks (CNNs)[18]

is quite other than this manner, the output of each convolutional layer fuses structural information along the range dimension and channel-wise information within local receptive fields instead of doing simple summations. Furthermore, the hierarchical patterns captured by CNNs lead to the invariance of translation, scaling, and shift, so that the features extracted by CNNs are more robust.[19] Therefore, CNNs and its variants have been successfully applied to solve the problems of HRRP-based RATR and outperform FC networks.[6,20-22]

However, CNNs have two shortcomings to be improved. On the one hand, the network structure of them used in HRRP-based RATR is generally designed by the rule of thumb, which needs extraordinary patience to carry out structure searching in a very time-consuming condition and always produce suboptimal network structure. On the other hand, CNN sare generally over-parameterized, that is, there are too many channels in each layer, which may lead to high computation complexity and overfitting.[23,24] As we all know, different channels can generate different feature maps, some are useful and the others are less useful or even superfluous, we should eliminate or weaken the less useful and redundant channels based on the Occam's razor principle.

In this paper, we try to find a relatively optimal structure to improve the performance of one-dimensional CNN (1D-CNN) for HRRP target recognition in an automatic manner. We accomplish this goal by two strategies, a.k.a., channel attention and channel pruning. Basically, we use 1D-CNN to extract different hierarchical features via multichannel stacked one-dimensional convolutional (conv1d) layers as well, but considering the interdependency between channels of different convolutional features, we embed an aggregation-perception-recalibration (APR) block into the 1D-CNN backbone for enhancing informative features while suppressing less useful ones. APR blocks consist ofmultiple FC layers which activated by Mish function. Furthermore, there are still some redundant channels in the networks that should be pruned to alleviate the computation complexity and overfitting. The lottery ticket hypothesis[25,26] states that there are smaller subnetworks (the winning tickets) within a larger network, which can be trained in isolation to achieve commensurate accuracy of the larger network. Liu et al.[27] and Lin et al.[28] further reveal that the best choice to acquire the winning tickets is to find the best number of channels in each layer. Similarly, to avoid the rule-of-thumb design of network structure, we employ an intelligent optimization algorithm, namely, the global best leading artificial bee colony (GBL-ABC)[29] algorithm to significantly prune the number of channels in 1D-CNN via automatic structure searching. GBL-ABC is an evolutionary algorithm for gradient-free optimization and it takes advantage of the updating equation of differential evolution algorithm in the employed bee phase and a global best leading strategy in the onlooker bee phase to accelerate convergence. In our implementations, network structure with a set of channel numbers in various network depth is regarded as a nectar source. Rather than randomly initiating the nectar sources, we first initiate the nectar sources in a relatively small spaceto make GBL-ABC search from a set of small network structures and progressively enlarge them according to their fitness. After that, the current best nectar source is introduced in the search equation of onlooker bee phase.

The proposed method includes the following advantages:

(1) Compared with other deep learning methods based on FC networks, our method utilizes 1D-CNN to capture the structural information among the range cells layer by layer, making it more robust to noise.

(2) The proposed APR block for channel attention utilizes MLP to adaptively learn the correlations between channels in each conv1d layer and then enhances the structural information of targets in range cells by layer-wise feature recalibrating, thus improving the recognition rate. The APR block has low computation complexity as well.
(3) The proposed GBL-ABC for channel pruning can not only reduce the computational complexity of 1D-CNN but also achieve the goal of improving the recognition ratefor HRRP recognition tasks by automatically searching for an optimal subnetwork from the original network, even though the pruning phase is relatively time-consuming.
(4) By introducing channel attention and channel pruning to 1D-CNN simultaneously, the proposed method achieves superior recognition performance and has good generalizability in some extent.

The rest of this paper is organized as follows. In Section 2, we first review related works and provide preliminaries. Then the structure of the proposed model and the GBL-ABC algorithm for automatic channel pruning are presented in detail in Section 3. In Section 4, several experiments are conducted on the simulated HRRP data set to evaluate our proposed method and we report experimental results. Finally, we draw concludes for our work in Section 5.

2 | PROBLEM FORMULATION

2.1 | HRRP-based target recognition

For an electrically large target, for example, the decoys and warheads in the midcourse of a ballistic missile, the target size of which is far larger than the wavelength of high-resolution radar, and the microwaves scatter at high frequencies,[30] so we can use the scattering center model to describe the electromagnetic characteristic of the target. As

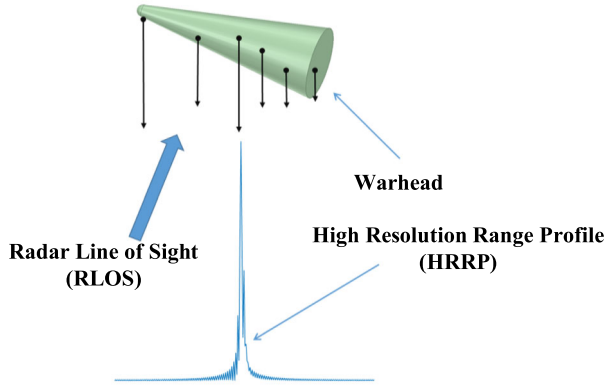


FIGURE 1 Illustration of a simulated HRRP sample from a warhead of a ballistic missile with scattering center model. HRRP, high-resolution range profile; RLOS, radar line-of-sight [Color figure can be viewed at wileyonlinelibrary.com]

shown in Figure 1, an HRRP sample consists of the amplitude of the coherent summations of the complex returns from target scatters in each range cell along RLOS. Concretely, the coherent summation of the complex returns of mth range cell in the nth HRRP sample can be described as follows:

$$x_n\left(m\right) = \sum_{i=1}^{V_m} \sigma_{m,i} exp\left[-j\left(\frac{4\pi R_{m,i}(n)}{\lambda}\right) + \varphi_{m,i}\right], \tag{1}$$

where V_m denotes the number of target scatters in the mth range cell, λ denotes the wavelength of the high-resolution radar, $\sigma_{m,i}$ and $\varphi_{m,i}$ represent strength and the initial phase of the ith scatter in the lth range cell, respectively, and $R_{m,i}(n)$ denotes the radial distance between the radar and ith range cell of the nth returned echo. Furthermore, for the nth returned echo with D range cells, we descript the corresponding HRRP sample, for the sake of simplicity as follows:

$$x_n = [x_n(1), x_n(2), ..., x_n(D)]. \tag{2}$$

Besides, to avoid the amplitude-scale sensitivity, the amplitude of each range cell in this paper is normalized as follows:

$$\bar{x}_n(i) = \frac{x_n(i)}{\max\limits_{j\in\{1,2,...,D\}}\{x_n(j)\}}. \tag{3}$$

For a data-driven radar HRRP target recognition task, which is also regarded as a cooperative target recognition,[1,5,6,10,11,15,17,20-22,31-38] we are provided with an independently and identically distributed set of HRRP samples $X = \{x_1, x_2, ..., x_N\}$, $x_i \in \mathbb{R}^D$ with corresponding true labels $Y = \{y_1, y_2, ..., y_N\}$, $y_i \in \{1,2, ..., S\}$ from different azimuths, N and S refers to the number of samples and classes, respectively. The deep learning method we used, a.k.a., CNN, is a kind of supervised learning and consists of two main stages, namely, the training stage and the evaluating stage. There are different data sets for each stage, we denote D_{train} and D_{test} as the training data set and evaluating data set. In the training stage, we use a criterion, a.k.a., loss function, to guide the training. We assume that the network is parameterized by θ, and the output of the network is $y_i' = f_\theta(x_i)$ with the input of x_i, we use empirical risk-likelihood as loss function, which is defined as follows:

$$J_\theta(X, Y) = \sum_{i=1}^{N} L(y_i, y_i'), \tag{4}$$

where $L(\cdot)$ is a nonnegative and observable function to quantify the distance between the true label y_i and predicted label y_i', $x_i \in D_{\text{train}}$. During the training stage, the network training is equivalent to learn parameters by minimizing the likelihood

$$\theta^* = \underset{\theta}{\text{argmin}}\, J_\theta(X, Y), \tag{5}$$

where θ^* is a set of trained parameters. As for the evaluating stage, we predict the input HRRP sample x_i by maximizing the posterior probability

$$\tilde{y}_i = \underset{k\in\{1,\ldots,S\}}{\operatorname{argmax}} P(y = k \mid x_i'; \theta) \tag{6}$$

where $\tilde{y}_i$ is the predicted label in the evaluating stage and $x_i' \in D_{\text{test}}$.

2.2 | Optimal 1D-CNN structure

Channel pruning is among the predominant approaches to reduce the computation complexity of model inference and widely used to find an optimal structure for a deep neural network.[39-41] The recent research reveals that the most important step of channel pruning is to find an optimal number of channels of each layer in CNN.[27,28] In this paper, we use a deep 1D-CNN interleaved with conv1d layers and channel attention blocks for HRRP recognition and try to find an optimal number of channels in an automatic manner.

For a 1D-CNN model N with L conv1d layers and its filterset $W = [W_1, W_2, \ldots, W_L]$, we denote $C = [c_1, c_2, \ldots, c_L]$ as the network structure of N, where c_j and W_j refer to the channel number and weights of the jth layer respectively. More concretely, $W_j = [w_{j1}, w_{j2}, \ldots, w_{jc_j}]$ and w_{ji} means the weights of the ith channel in the jth layer. Given any subnetwork N' pruned from N, we denote $C' = [c_1', c_2', \ldots, c_L']$ and $W' = [W_1', W_2', \ldots, W_L']$ as the network structure and filter set of N' respectively, obviously $c_j' \le c_j$. In this paper, the aim of pruning is finding a subnetwork N' with an optimal combination C' trained on D_{train} to achieve the best accuracy on D_{test}. To this end, we formulate the problem of pruning 1D-CNN as follows:

$$(N')^* = \underset{N'}{\operatorname{argmax}}\, acc(N'(C', W'; D_{\text{train}}); D_{\text{test}}), \tag{7}$$

where $acc(\cdot)$ denotes the evaluating accuracy on D_{test} for the pruned model N' with structure C'. The evaluating accuracy is defined as the ratio of correctly classified samples to the total tested samples.

3 | METHOD

The framework of the proposed method 1D-CNN with channel attention and channel pruning (CNN1D-CACP) for HRRP recognition is illustrated in Figure 2. As mentioned above, our method includes two stages, a.k.a., training stage and evaluating stage. There are three procedures in the training stage, namely pretraining, pruning, and fine-tuning, we will talk about them for detail in Section 3.4. In the evaluating stage, we utilize a softmax classifier for classification and evaluation.

3.1 | 1D-CNN

A deep CNN usually stacks a series of convolutional layers with nonlinear activation functions and down-sampling pooling layers, so that it can extract hierarchical features with theoretically global receptive fields. A kin to this manner, we take advantage of deep 1D-CNN to capture hierarchical features of HRRP for classification and the basic stacked

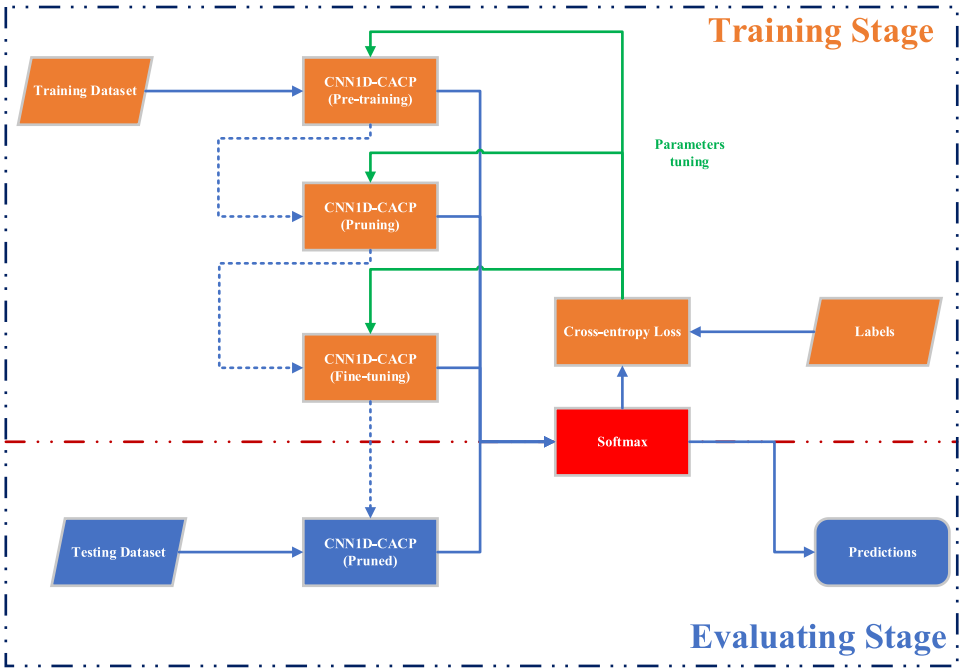


FIGURE 2 Schematic diagram of the proposed method [Color figure can be viewed at wileyonlinelibrary.com]

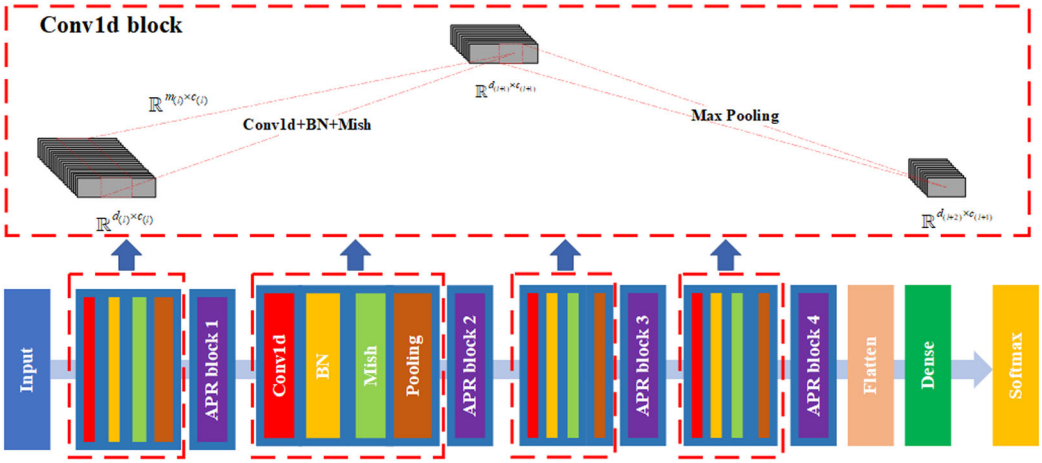


FIGURE 3 The model structure of CNN 1D-CACP. A series of conv1d and APR blocks are interleaved to extract hierarchical features, and a softmax classifier receives the hierarchical features and outputs the classification results. APR, aggregation-perception-recalibration; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; conv1d, one-dimensional convolutional [Color figure can be viewed at wileyonlinelibrary.com]

block of deep 1D-CNN, termed as conv1d block in this paper, is shown in Figure 3. The conv1d block consists of three operations, namely, conv1d, batch normalization, and pooling operation.

3.1.1 | 1D convolution

Suppose $X^{(l)} \in \mathbb{R}^{d_{(l)} \times c_{(l)}}$ is a multichannel input of the lth conv1d layer, where $d_{(l)}$ represents the dimensionality of feature maps and $c_{(l)}$ denotes the number of channels in the lth layer. For kth filter at the lth layer, $k \in \{1,2, ..., c_l\}$, a conv1d operation with stride length $r^{(l)}$ applies a filter $W^{(k,l)} \in \mathbb{R}^{m_{(l)} \times c_{(l)}}$ to $X^{(l)}$. Thus each channel yields a feature map

$$X^{(k,l+1)} = \delta(X^{(l)} * W^{(k,l)} + b^{(k,l)}), \tag{8}$$

where $X^{(k,l+1)} = \delta(X^{(l)} * W^{(k,l)} + b^{(k,l)})$, denotes the convolution operator, $b^{(k,l)}$ denotes the bias for kth feature map, and $\delta(\cdot)$ is a nonlinear activation function. In this paper, Mish[42] function is applied to the 1D-CNN as the activation function, which is defined as follows:

$$\delta(x) = x \cdot \tan h(\varsigma(x)), \tag{9}$$

where $\varsigma(x) = \ln(1 + e^x)$ refers to the softplus[43] activation. More specially, the value of the conv1d operation at position i is given by

$$X_i^{(k,l+1)} = \delta\left(\sum_{j=1}^{m_{(l)}} \sum_{c=1}^{c_{(l)}} X^{(l)}_{r^{(l)} \times (i-1)+1+j,c} \odot W^{(k,l)}_{j,c} + b^{(k,l)}\right), \tag{10}$$

where $\odot$ refers to Hadamard product and $m_{(l)}$ is the window size of filters in the lth conv1d layer. Accordingly, the dimensionality of the output feature map $X^{(k,l+1)}$ after conv1d operation is

$$d_{(l+1)} = \left\lfloor \frac{d_{(l)} - m_{(l)} + 2 \times p_{(l)}}{r^{(l)}} \right\rfloor + 1, \tag{11}$$

where $p_{(l)}$ is the amount of implicit zero-paddings on both sides in the input feature map for padding number of points. From Equation (10), we can see that all conv1d operations in each channel share the same conv1d filter, which can fuse structural information along the range dimension of HRRP samples and channel-wise information within local receptive fields.

3.1.2 | Batch normalization

Batch normalization (BN)[44] is widely deployed in a deep neural network to accelerate training by reducing internal covariateshift and improve generalization by regulating the distribution of the inputs in each layer. We deploy batch normalization after each conv1d layer to make each layerhave a mean of 0 and a variance of 1. Suppose the output of each conv1d layer is $x = (x^{(1)}, x^{(2)}, ..., x^{(d)})$, we will normalize each dimension as follows:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}] + \varepsilon}}, \tag{12}$$

where the expectation and variance are calculated over the mini-batch from the training data set and ε is a constant added to the mini-batch variance for numerical stability. After that, two hyper-parameters, $\gamma^{(k)}$ and $\beta^{(k)}$, are introduced to scale and shift the normalized value:

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}. \tag{13}$$

3.1.3 | Pooling

The pooling layer can downsample the feature map, reducing dimensionality and retaining useful information layer by layer, the max-pooling is adopted in this paper. For the kth input feature map $X^{(k,l)}$ in the lth pooling layer, the value of max-pooling operation at the position i can be calculated by

$$X_i^{(k,l+1)} = \max_{j\in\{1,2,\ldots,w_{(l)}\}} \left\{ X_{r^{(l)}\times(i-1)+1+j}^{(k,l)} \right\}, \tag{14}$$

where $w_{(l)}$ is the window size of the pooling operator. Therefore, the dimensionality of the output feature map $X^{(k,l+1)}$ after pooling operation is

$$d_{(l+1)} = \left\lfloor \frac{d_{(l)} - w_{(l)}}{r^{(l)}} \right\rfloor + 1. \tag{15}$$

3.2 | Channel attention

Different channels attain different feature maps in the conv1d block, traditional deep models take into consideration those features of equal importance independently. In fact,

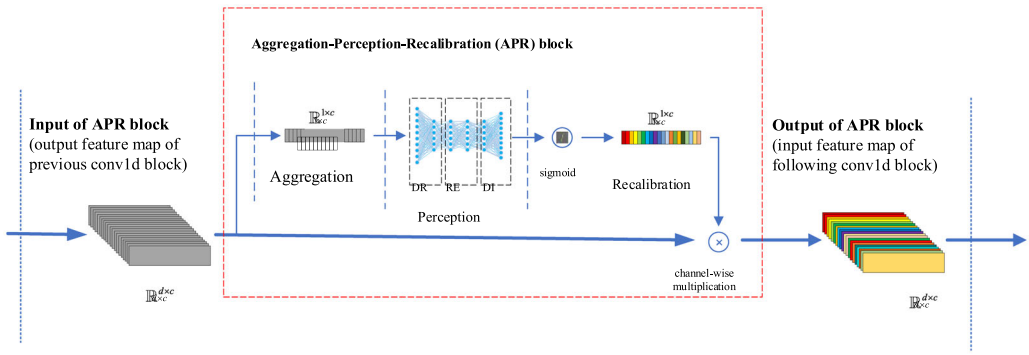


FIGURE 4 Illustration of APR block. APR, aggregation-perception-recalibration [Color figure can be viewed at wileyonlinelibrary.com]

as different feature map reflects unique structural information of targets in HRRP, there exits different importance between them.[45] Hence, the features should be recalibrated to achieve the goal of improving the quality of representations. To this end, we impose an APR block to the tail of the conv1d block, through which it is capable of learning to make use of global information to selectively and adaptively highlight representative features and suppress less informative ones.

Suppose the output of lth conv1d layer is $X^{(l+1)} \in \mathbb{R}^{d_{(l+1)} \times c_{(l+1)}}$, where $d_{(l+1)}$ is calculated by Equation (15), for simplicity, $X^{(l+1)}$ is replaced by χ. As shown in Figure 4, the APR block which consists of three operations, including aggregation, perception, and recalibration, is embedded into the tail of the conv1d block.

3.2.1 | Aggregation

The mission for APR block is to figure out the correlations between different channels, to this end, the output feature χ of the anterior conv1d block is exploited. In consideration of that χ is generated by conv1d block within local receptive fields, so that it does not have enough global spatial information outside of this region in a corresponding feature map. To tackle this problem, we aggregate the global spatial information outside of the local receptive fields by a global max-pooling operation. For cth channel, we can generate a statistic by

$$z_c = F_{agg}(\chi_c) = \max_{i \in \{1,2,\ldots,d\}} \{\chi_c(i)\}, \tag{16}$$

where d means the dimensionality of the cth feature mapχ_c, z_c is an aggregated statistic expressive for cth channel. By doing this, we manage to acquire a global statistic, that is, the max value in this paper, of the whole feature map in the corresponding channel.

3.2.2 | Perception

To perceive the dependencies between different channels based on the statistics gained from the aggregation operation, we take advantage of MLP to deal with it. Perception operation can learn a nonlinear relationship between different channels with MLP. We further divide the MLP into three layers for the convenience of parameter setting description, these three layers include dimensionality-reduction (DR) layer with a reduction ratio μ, a representative enhancing (RE) layer and a dimensionality-increasing (DI) layer. The perception operation can be described as follows:

$$u = F_{per}(z, W) = W_{DI}\delta(W_{RE}\delta(W_{DR}z)), \tag{17}$$

where δ means Mish activation function, $W_{DR} \in \mathbb{R}^{\frac{C}{\mu} \times C}$, $W_{RE} \in \mathbb{R}^{\frac{C}{\mu} \times \frac{C}{\mu}}$, and $W_{DI} \in \mathbb{R}^{C \times \frac{C}{\mu}}$. The DI layer returns to the channel dimension of global statistics z. There is a series of FC layers in the RE layer sharing the same number of parameters to enhance the representation of the interaction between channels, and the number of FC layers in the RE layer is parameterized by η. The parameter choice of η as well as μ will be discussed in

Section 4.3. We opt to deploy the Mish activation function between every two FC layers as well.

3.2.3 | Recalibration

To recalibrate features extracted by each conv1d block, a simple gating mechanism with a sigmoid activation to the output of the perception operation is first employed. After that, the final output of the APR block is obtained by rescaling χ

$$\tilde{\chi}_c = F_{rec}(\chi_c, \sigma(u_c)) = \chi_c \sigma(u_c), \tag{18}$$

where σ refers to the sigmoid activation function, F_{rec} refers to channel-wise multiplication and $\tilde{\chi} = [\tilde{\chi}_1, \tilde{\chi}_2, ..., \tilde{\chi}_C]$.

The APR block transforms the original output of the conv1d block to a set of weights which is tightly related to different channels and can be dynamically adjusted during training. Since the recalibration of each channel depends on its correlation to the others, the APR block can be deemed to be a kind of self-attention mechanism. On the existence of the LR layer, the APR block can learn how to recalibrate the channels by perceiving its latent relationship, thus achieving the goal of adaptively channel calibrating.

3.3 | Channel pruning

For a network N with a structure of C, there are $\prod_{i=1}^{L} c_i$ candidates for the pruned network, walking through these candidates can be very time-consuming, thus shrinking the combinations of the network should be considered to limit structure searching. At the same time, the number of network connections after pruning should be as small as possible based on the principle of Occam's Razor. To that effect, we consider imposing constrain to Equation (7)

$$\begin{aligned} (N')^* = & \underset{N'}{\operatorname{argmax}}\, acc(N'(C', W'; D_{\text{train}}); D_{\text{test}}) \\ & \text{s. t. } c_i' \le \alpha c_i,\ i \in \{1, 2, ..., L\} \\ & W' \subset W \end{aligned} \tag{19}$$

where $\alpha \in (0,100\%]$, is a pre-given max threshold proportion of the preserved channels in the pruned model N', when $\alpha c_i < 1$, we set c_i' to preserve 1 channel. To avoid randomly reinitializing the weights of N' to reduce computation cost, the weights of N' are directly inherited from N, that is, $W' \subset W$. With the hyper-parameter α, the candidates for searching can be significantly decreased to $\prod_{i=1}^{L} \alpha c_i$. Thus, the aim of pruning can be transformed into finding an optimal channel number c_i', which is not more than αc_i in ith layer, to maintain or exceed the performance of the pretrained model. Furthermore, to avoid enumerating each candidate in the pretrained network, we consider integrating a heuristic algorithm, that is, GBL-ABC, to prune the pretrained network by searching for an optimal structure automatically. More evocatively, the network structure C is deemed to be a nectar source in GBL-ABC.

Algorithm 1. Pruning a pretrained network by GBL-ABC.

Input: Pretrained network: N, Epochs: T, Number of nectar sources: K, Max threshold proportion: α, Max times for nectar sources to stop updating: M, Training data set: D_{train}, Testing data set: D_{test}, Training epochs for calculating fitness: Ω.

Output: Optimal pruned network: $(N')^*$.

Set counters to zero: $\{t_j\}_{j=1}^{K} = 0$;
Initialize the nectar source set $\{C_j'\}_{j=1}^{K}$, for each of them, randomly select a subnetwork according to C_j' as a pruned network N_j' with its filter set W_j'; via Algorithm 2;
for $i = 1 \rightarrow T$ **do**
  for $j = 1 \rightarrow K$ **do**
    generate a new nectar source G_j' via Equation (20);
    select a subnetwork from Naccording to G_j' with its filter set W_j' generated via Algorithm 2;
    training C_j' and G_j' for Ω epochs to calculate the fitness of them via Equation (21);
    if $Fit_{C_j'} < Fit_{G_j'}$ **then**
      $C_j' = C_j'$;
      $Fit_{C_j'} = Fit_{G_j'}$;
      $t_j = 0$;
    else
      $t_j = t_j + 1$;
    end
end
  for $j = 1 \rightarrow K$ **do**
    calculate P_j via Equation (23)
    generate a random number $\varepsilon_j \in [0,1]$;
    if $\varepsilon_i <= P_j$ **then**
      generate a new nectar source G_j' via Equation (22);
      randomly select a subnetwork from N according to G_j' with its filter set W_j' generated via Algorithm 2;
      traning C_j' and G_j' for Ω epochs to calculate the fitness of them via Equation (21);
      if $Fit_{C_j'} < Fit_{G_j'}$ **then**
        $C_j' = C_j'$;
        $Fit_{C_j'} = Fit_{G_j'}$;
        $t_j = 0$;
      else
        $t_j = t_j + 1$;
      end
    end
end
for $j = 1 \rightarrow K$ **do**
  if $t_j > M$ **then**
    reinitialize C_j';
    randomly select a subnetwork from N according to C_j' with its filter set W_j' generated via Algorithm 2;
  end
 end
end
return $(N')^* = \underset{N_j'}{argmax}\ acc(N_j'(C_j', W_j'; D_{train}); D_{test})$;

Algorithm 2. *Generating a subnetwork from a pretrained network randomly.*

Input: Filter set of the pretrained network N: W, Structure of the subnetwork N': $C' = [c_1', c_2', ..., c_L']$.
Output: The filter set of the subnetwork N': W'.

1: **for** $l = 1 \rightarrow L$ **do**
2: $z = c_l'$
3: get the filter number of the lth conv1d layer: $n_1 = len(W_l)$;
4: generate a one-dimensional array $A = [a_1, a_2, ..., a_z]$ of positive integers and the ith element a_i is randomly selected form $[1, n_1]$, where $a_1 \neq a_2 ... \neq a_z$;
5: sort the array A from small to large;
6: **for** $i = 1 \rightarrow c_l'$ **do**
7: $j = a_i$;
8: $w_{l_i}' = w_{lj}$;
9: end
10: end
11: return W'

The pseudocode of the proposed method for automatically channel pruning is described in Algorithm 1, which is further refined into four phases below.

3.3.1 | Initialization phase (Line 1–2)

In this phase, a set of nectar sources that represent K candidates of the pruned structure are denoted by $\{C_j'\}_{j=1}^{K}$, and ith element c_{ji}' of nectar source C_j' is randomly selected from $[1, \frac{\alpha c_i}{\tau}]$, where $\tau \in [1, +\infty)$. An upbound $\frac{\alpha c_i}{\tau}$ is introduced for making GBL-ABC search from a set of small network structures and progressively enlarge them according to their fitness. For each candidate of the pruned structure, the weights of the corresponding pruned network are directly inherited from the pretrained network via Algorithm 2.

3.3.2 | Employed bee phase (Line 4–15)

An employed bee hunts a new nectar source G_j' for each nectar source C_j'. In particular, the ith element of G_j' is generated from three randomly selected neighbor elements. The search equation, which is originated from the differential evolution algorithm, is denoted as follows:

$$g_{ji}' = c_{ai}' + F_i(c_{bi}' - c_{ci}') \tag{20}$$

where $a \neq b \neq c \neq j$, F_i is a mutation scale related to each element. In this paper, F_i is dependently generated by a Gaussian distribution with a mean value of 0.5 and a standard deviation of 0.1.

After producing a new nectar resource, the GBL-ABC applies a greedy selection principle between nectar source C_j' and G_j' according to their fitness, which is defined as below

$$Fit_{C_j'} = acc(N_j'(C_j', W_j'; D_{\text{train}}); D_{\text{test}}). \tag{21}$$

3.3.3 | Onlooker bee phase (Line 16–31)

In this phase, we use the global best position of nectar source to act as a guide for the onlooker bees to update $\{C'_j\}_{j=1}^{n}$, to replace the Equation (20), the search equation is updated as follows:

$$g'_{ji} = gbest_{ji} + M_i(c'_{ai} - c'_{bi}) \tag{22}$$

where $gbest_{ji}$ is the current global best position, c'_{ai} and c'_{bi} are randomly selected neighbors of c'_{ji} and $a \neq b \neq j$, and M_i is generated by Gaussian distribution with a mean value of 0.0 and a *standard deviation* of 2.0. Besides, a nectar source C'_j is chosen with a probability related to its fitness via a roulette wheel selection strategy. This probability in our implementation is defined as follows:

$$P_j = 0.85 \cdot \frac{Fit_{C'_j}}{\max\limits_{j \in \{1,2,\ldots,K\}} (Fit_{C'_j})} + 0.15. \tag{23}$$

Obviously, the better fitness of C'_j is, the higher probability of C'_j being selected. This will make the algorithm find an optimal nectar source automatically and progressively.

3.3.4 | Scout bee phase (Line 32–38)

A scout bee will prevent the algorithm from being trapped in a local optimal. On condition that the nectar source C'_j has not been updated more than M times, the scout bee will reinitialize it to further generate a new nectar source.

3.4 | Construction of the proposed model

As shown in Figure 3, the proposed method CNN1D-CACP uses four stacked conv1d-APR blocks to extract hierarchical features of HRRP samples, that is, $L = 4$. Afterward, a Flatten operation is imposed to covert the multidimensional tensor of hierarchical features into a 1D vector, which is suitable for the input of the following softmax classifier. Given an input HRRP sample x_i, the softmax function is adopted to predict the label vector, which is

$$h_\theta(x_i) = \begin{bmatrix} P(y=1|x_i;\theta) \\ P(y=2|x_i;\theta) \\ \vdots \\ P(y=S|x_i;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{S} exp(\theta_j^T x_i)} \begin{bmatrix} exp(\theta_1^T x_i) \\ exp(\theta_2^T x_i) \\ \vdots \\ exp(\theta_S^T x_i) \end{bmatrix} \tag{24}$$

where $\theta = \left[\theta_1^T, \theta_2^T, \ldots, \theta_S^T\right]^T$ represents all the parameters to be trained in the network. In the evaluating stage, we use Equation (6) for predictions.

As for the training stage, we adopt the cross-entropy loss as the cost function,

$$J_\theta(X, Y) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{S} 1\{y_i = k\} log \frac{exp\left(\theta_j^T x_i\right)}{\sum_{j=1}^{s} exp\left(\theta_j^T x_i\right)}\right] \tag{25}$$

where m is the sample number of each mini-batch and $1\{\cdot\}$ is an indicator function with $1\{\text{true}\} = 1$ and $1\{\text{false}\} = 0$.

As shown in Figure 2, the training stage is divided into three phases:

3.4.1 | Pretraining phase

We first initialize the network N by Kaiming initialization[46] and then train it for several epochs to conduct pruning. The RAdam[47] optimizer, a state-of-the-art advanced mini-batch stochastic gradient descent algorithm, is further incorporated for training according to Equation (5).

3.4.2 | Pruning phase

Note that the fitness used by GBL-ABC algorithm refers to the evaluating accuracy on the test data set, we also train the pruned model N' for several epochs to calculate the fitness with RAdam. As elaborated in Algorithm 2, to avoid training subnetwork from scratch, we randomly pick up c_j' filters from the pretrained model N, which serve as the initialization for the jth conv1d layer of the subnetwork N'.

3.4.3 | Fine-tuning phase

To achieve better accuracy for the pruned model, we lastly fine-tune the pruned model for more epochs using RAdam as well.

As mentioned above, all pruned networks in the four phases of the pruning phase are trained for Ω epochs to calculate their fitness via Equation (21), we can estimate the lower bound of the extra training epochs during the pruning phase to analyze the complexity of GBL-ABC. Since each nectar source represents a pruned network, the extra training epochs in the initialization phase, employed bee phase, and onlooker bee phase are all $K \times \Omega$ during one cycle of the pruning phase, the extra training epoch in the scout bee phase is hard to calculate precisely, we use $E^i{}_{\text{scout}}$, $i \in \{1,2, ..., T\}$ to denote it. As a whole, the total extra training epoch in the pruning phase is $\sum_{i=1}^{T}(3 \times K \times \Omega + E^i_{\text{scout}})$, hence we should train equivalently at least for $T \times 3 \times K \times \Omega$ epochs during the pruning phase, which is relatively time-consuming. For deep learning models, reducing the computational complexity of the modelto save inference time in the evaluating stage is more important. As we introduced a hyper-parameter α to control the maximum number of channels of the pruned networks, the model parameters can be reduced according to the lottery ticket hypothesis, thus reducing the computational complexity of the model.

4 | RESULTS AND DISCUSSION

4.1 | Measured data

We use an electromagnetic simulation software, a.k.a., FEKO, to simulate radar echoes from five mid-course ballistic targets, that is, $S = 5$, and the physic characteristics of these targets are shown in Figure 5. Based on the high-frequency asymptotic theory, the physical optics method is adopted for simulation.[48] The FEKO simulation parameters are set as follows: azimuth angle is 0–180°, angle step is 0.05°, the pitch angle is 0, the center frequency is 10 GHz with start frequency 9.5 GHz and end frequency 10.5 GHz and the number of frequency sampling points is 128. At last, the default optimal mesh size and horizontal polarization are adopted.

Finally, there are 18,005 samples attained through processing by inverse fast Fourier transform,[49] each target contains 3601 HRRP samples of different degrees and each sample is a 256-dimensional vector, that is, $D = 256$. We randomly split these samples into two data sets, the testing data set with 3601 samples and the training data set with 14,404 samples, each data set contains the same number of samples for different targets. The HRRP simulated results of different targets with the azimuth of 90° are illustrated in Figure 6.

4.2 | Model settings

To investigate the effect of incorporating 1D CNN with channel attention and channel pruning for HRRP recognition, we compare the proposed method CNN1D-CACP against several existing models for HRRP-based RATR. These models all contain deep networks:

ML-ELM: multilayer extreme learning machine.[50] SAE-ELM: stacked auto-encoders and extreme learning machine.[51] SAE: stacked auto-encoders.[52] SSpAE: stacked sparse auto-encoders.[5] SDAE: stacked denoising auto-encoders.[53] Bi-GRU: bidirectional GRU.[17] Bi-LSTM: bidirectional LSTM.[54] CNN1D: the vanilla 1D-CNN, which is not incorporated with channel attention or channel pruning. CNN1D-CA: the 1D-CNN with channel attention. CNN1D-CP: the 1D-CNN with channel pruning.

Except for ML-ELMand SAE-ELM, all neural networks mentioned above are trained to minimize the cross-entropy loss for 200 epochs, and the softmax classifier is adopted in these networks for classification as well. We use RAdam for minimizing the cross-entropy loss with 32 samples in each mini-batch, a learning rate of 10^{-3} and standard hyper-parameter values $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$, the learning rate is divided by 10 every 30 epochs. For the proposed model, we set $m_{(l)} = 5$, $p_{(l)} = 0$, $r^{(l)} = 1$ in the conv1d layer and $w_{(l)} = 3$, $r^{(l)} = 3$ in the pooling layer, where $l \in \{1,2,3,4\}$. For Algorithm 1, we set $\tau = 2$, $T = 20$, $K = 10$, $M = 5$, $\Omega = 2$, and all optimal pruned networks are fine-tuned for 200 epochs.

All models are implemented withdeep learning framework PyTorch[55] 1.15.1 and programing language Python 3.7 on an x86-64 Fedora PC, which is based on Intel Core i5-9600K CPU @3.7 GHz, 24 G RAM, and NVIDIA GTX 1060 GPU with CUDA 10.2 accelerating computation.

4.3 | Impact of model hyper-parameters

In this section, we conduct a series of experiments to analyze the influence of hyper-parameters on the performance for choosing suitable network settings of the proposed method.

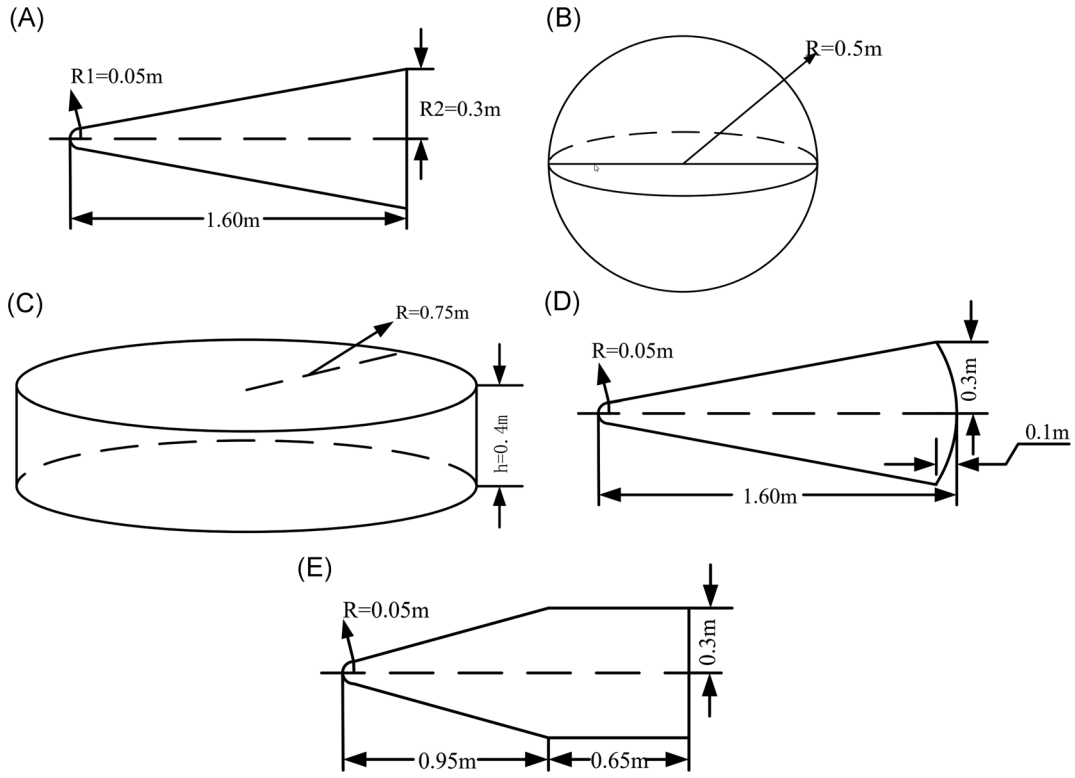


FIGURE 5 The physic characteristics of five mid-course ballistic targets. (A) Warhead. (B) Decoy 1. (C) Decoy 2. (D) Decoy 3. (E) Decoy 4

The performance metrics for measuring the network compressing in this paper include the total channel number, floating-point operations (FLOPs), and the parameter number of the model. The average evaluating accuracy (Acc) of different models is also provided. We provide ablation studies for the following hyper-parameters:

4.3.1 | The influence of C

To evaluate the efficiency of the proposed method in different 1D CNN model configurations, the channel number of the pretrained model in each conv1d layer, which is represented by $C = [c_1, c_2, c_3, c_4]$, is considered. More specifically, we take into account five model configurations, that is, $C_1 = [20, 40, 80, 160]$, $C_2 = [40, 80, 160, 320]$, $C_3 = [60, 120, 240, 480]$, $C_4 = [80, 160, 320, 640]$, and $C_5 = [100, 200, 400, 800]$, and four model architectures, that is, CNN1D, CNN1D-CP, CNN1D-CA, and CNN1D-CACP. Table 1 depicts both evaluating accuracy and channel number of the models in different architectures and configurations. As shown in Table 1, channel pruning and channel attention consistently improve performance in different model configurations, respectively, we see further increasing of the accuracy when channel pruning and channel attention are combined simultaneously, which proves the effectiveness of the proposed method.

Moreover, we plot the training and validation curves of the five architectures with the model configuration of C_5 in Figure 7. From Figure 7, we can see that CNN1D-CA, CNN1D-CP,

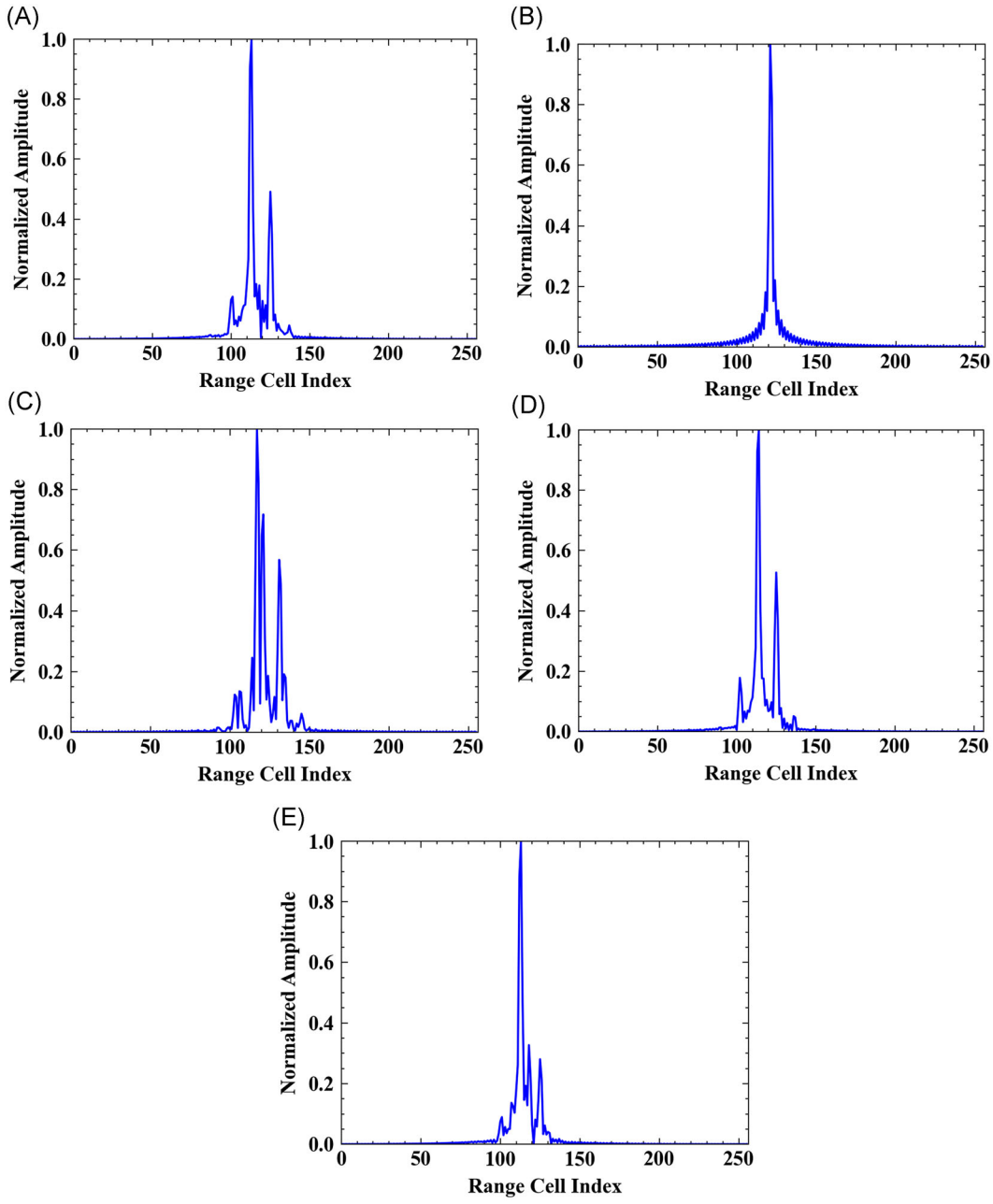


FIGURE 6 The simulated HRRP samples of five different mid-course ballistic targets with the azimuth of 90°, which are all normalized by Equation (3). There are 256 range cells in each HRRP sample and all are arranged along RLOS. (A) Warhead. (B) Decoy 1. (C) Decoy 2. (D) Decoy 3. (E) Decoy 4. HRRP, high-resolution range profile; RLOS, radar line-of-sight [Color figure can be viewed at wileyonlinelibrary.com]

and CNN-CACP consistently surpass CNN1D during training in both training data set and testing data set, and CNN1D-CACP shows the best performance. Since the channel number has been reduced after pruning, it proves the evidence that GBL-ABC can get rid of the redundancies which lead to overfitting. Figure 8 shows the Giga FLOPs (GFLOPs) and the

TABLE 1 The evaluating accuracy and channel number of the model in different architectures and configurations

Network configuration	CNN1D		CNN1D-CP		CNN1D-CA		CNN1D-CACP	
	Acc (%)	Channels	Acc (%)	Channels	Acc (%)	Channels	Acc (%)	Channels
C_1	96.00	300	97.75	136	96.31	300	97.83	137
C_2	96.56	600	97.89	211	96.70	600	98.11	408
C_3	96.66	900	98.22	353	96.68	900	98.36	278
C_4	96.70	1200	98.20	694	96.89	1200	98.25	742
C_5	96.69	1500	98.06	377	96.83	1500	98.42	699

Abbreviations: 1D, one-dimensional; Acc, accuracy; CNN, convolutional neural network; CNN1D, one-dimensional CNN; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CP, 1D-CNN with channel pruning; CNN1D-CACP, 1D-CNN with channel attention and channel pruning.

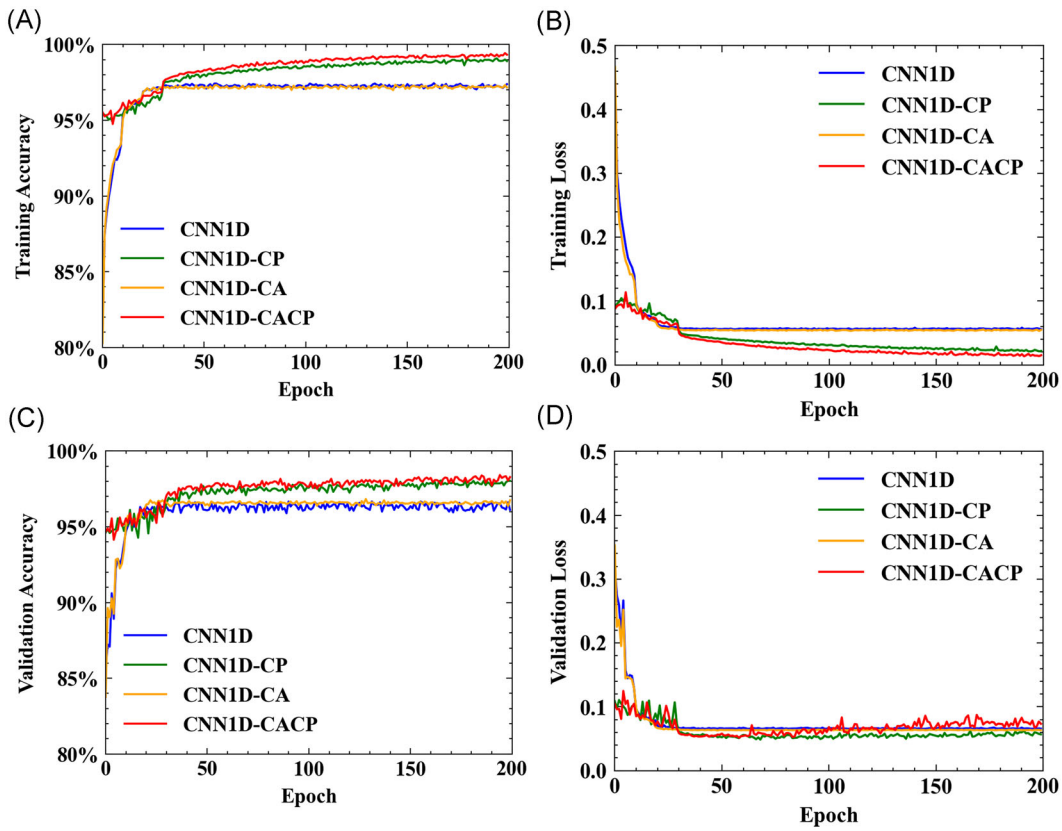


FIGURE 7 The training and validation curves of CNN1D, CNN1D-CA, CNN1D-CP, and CNN-CACP with the model configuration of C_5. (A) Training accuracy curves. (B) Training loss curves. (C) Validation accuracy curves. (D) Validation loss curves. 1D, one-dimensional; CNN, convolutional neural network; CNN1D, one-dimensional CNN; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CP, 1D-CNN with channel pruning; CNN1D-CACP, 1D-CNN with channel attention and channel pruning [Color figure can be viewed at wileyonlinelibrary.com]

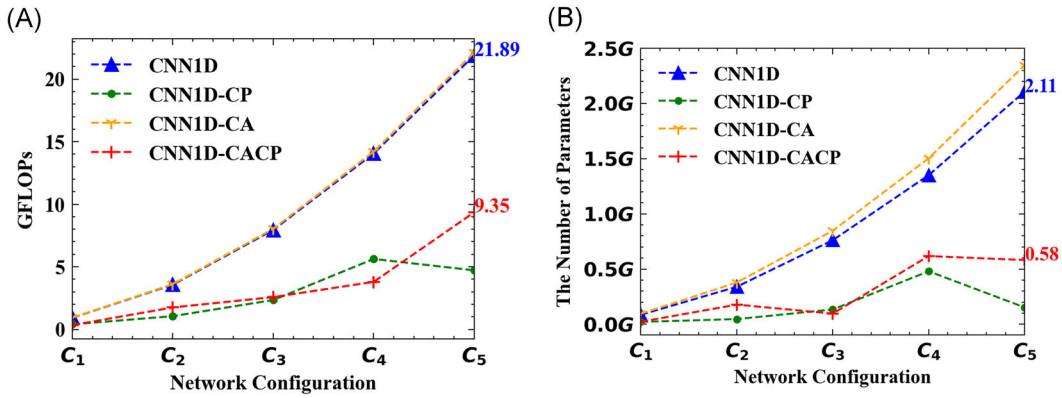


FIGURE 8 The GFLOPs and the parameter number of the model in different architectures and configurations. (A) GFLOPs. (B) The number of parameters. 1D, one-dimensional; CNN, convolutional neural network; CNN1D, one-dimensional CNN; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CP, 1D-CNN with channel pruning; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; GFLOP, Giga floating-point operation [Color figure can be viewed at wileyonlinelibrary.com]

parameter number of the model in different architectures and configurations. From Figure 8, we have two observations. On one hand, compared with the vanilla CNN1D, the GFLOPs and parameter numbers are increasing slightly in CNN1D-CA, proving the evidence that the APR block proposed in this paper has a small additional computational cost. On the other hand, compared with vanilla CNN1D and CNN1D-CA, the GFLOPs and parameter number of CNN1D-CP and CNN1D-CACP are decreased more obviously after pruning as the channel number in different pretrained model increases, proving that model with more channels has more redundancies and the proposed method can automatically find and remove the pointless parameters.

Based on the phenomenon above, we can also find that the proposed model CNN1D-CACP is capable of consistently improving evaluating accuracy of vanilla CNN1D while reducing the computational complexity under different network configurations with the help of channel attention mechanism to recalibrate learned features and channel pruning mechanism to find optimal network structure. To some extent, it proves that our solution has good generalizability on different network configurations.

Since CNN1D-CACP achieves the best evaluating accuracy with the configuration of C_5, we use C_5 as the default model configuration for all experiments. Under the configuration of C_5, CNN1D-CACP has significantly reduced the channel number, the computational complexity and the parameter number to $699/1500 \approx 46.6\%$,$9.35/21.89 \approx 42.71\%$ and $0.58/2.11 \approx 27.49\%$ of that of CNN1D respectively, even though it has been equivalently trained for at least $T \times 3 \times K \times \Omega = 20 \times 3 \times 10 \times 2 = 1200$ extra epochs in the pruning phase.

4.3.2 | The influence of η

With the model configuration of C_5, we analyze the influence of the introduced hyperparameter η, which represents the number of FC layers in the RE layer. We conduct experiments for a range of different η values and the results are shown in Table 2. As can be seen

TABLE 2 The influence of the FC layer number in the RE layer

	CNN1D-CA				CNN1D-CACP			
η	Acc (%)	FLOPS (k)	Parameters (k)	Channels	Acc (%)	FLOPs (k)	Parameters (k)	Channels
0	96.83	22099.60	2321.41	1500	98.17	3762.52	406.35	604
1	96.31	22112.87	2334.67	1500	98.39	5827.51	336.71	529
2	96.83	22126.14	2347.94	1500	98.42	9346.78	581.95	699
3	96.28	22139.48	2361.21	1500	98.22	3384.32	289.34	524
4	96.45	22152.68	2374.48	1500	97.97	4897.83	344.50	539

Abbreviations: Acc, accuracy; 1D, one-dimensional; CNN, convolutional neural network; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; FLOP, floating-point operation.

fromTable 2, FLOPs and the parameters increase along with η and achieve the best accuracy when $\eta = 0$ and $\eta = 2$ for the nonpruned models. In the pruned models, the accuracy varies from increasing to decreasing and achieve the best when $\eta = 2$. Since pruned models all have less computation complexity than nonpruned ones, we just focus on getting the best accuracy of the pruned model, so that we set $\eta = 2$ in this paper.

4.3.3 | The influence of μ

Table 3 depicts the accuracy and model complexity with different values of μ, which represent the reduction ratio of the APR block. From Table 3, it is clear that larger μ leads to increase the computation complexity and slightly decreases the accuracy of the nonpruned models. The proposed method can automatically find the best pruned structure as well and setting $\mu = 8$ achieves the best accuracy of the pruned models in this paper. Therefore, we set $\mu = 8$ as default.

4.3.4 | The influence of α

The max threshold proportion α is introduced in Equation (19) as a hyper-parameter which allows us to constrain the proportion of the preserved channels during channel pruning. To investigate the influence made by this hyper-parameter, we conduct a series of experiments with the default settings mentioned above for α varying from 10% to 100%, the comparison is showing in Figure 9. From Figure 9, we can see that larger αleads to less reduction of computation complexity and there are small fluctuations of the evaluating accuracy while increasing α. When $\alpha = 70\%$, the model achieves the best accuracy and we use this value of α in the following experiments. For a deeper analysis of channel pruning, we display block-wise pruning results of CNN1D-CP (pruned from CNN1D) and CNN1D-CACP (pruned from CNN1D-CA) in Figure 10. As shown in Figure 10, we can see that the pruning rates differ across different blocks when the max ratio of the preserved channels in each conv1d layer is set as the same value of 70%, proving the evidence that the proposed method can automatically find the best structure which can promote the evaluating accuracy.

TABLE 3 The influence of the reduction ratio

	CNN1D-CA				CNN1D-CACP			
μ	Acc	FLOPS (k)	Parameters (k)	Channels	Acc	FLOPs (k)	Parameters (k)	Channels
4	96.83	22418.45	2640.26	1500	98.25	4308.10	279.81	485
8	96.83	22126.14	2347.94	1500	98.42	9346.78	581.95	699
16	96.42	21999.81	2221.62	1500	97.86	1342.87	160.17	434
32	96.53	21941.43	2163.23	1500	97.67	3484.10	146.13	364

Abbreviations: Acc, accuracy; 1D, one-dimensional; CNN, convolutional neural network; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; FLOP, floating-point operation.

4.4 | Recognition performance

In this section, the proposed method is compared with other deep neural methods mentioned in Section 4.2 and the compared results are depicted in Table 4. As can be seen from Table 4, all deep neural networks can recognize all samples of decoy 1 in the test data set correctly, but CNNs has the superior recognition ratethan the other four targets. Since CNNs can capture structural information among range cells better than fully connected networks, CNNsalso outperform fully connected networks in total accuracy. In contrast to vanilla CNN1D, the embedded APR block for channel attention (CNN1D-CA) mainly improves the recognition rate of warhead targets, the improvement of GBL-ABC for channel pruning (CNN1D-CP) mainly comes from the improvements of the recognition rate of decoy 2 and warhead targets.

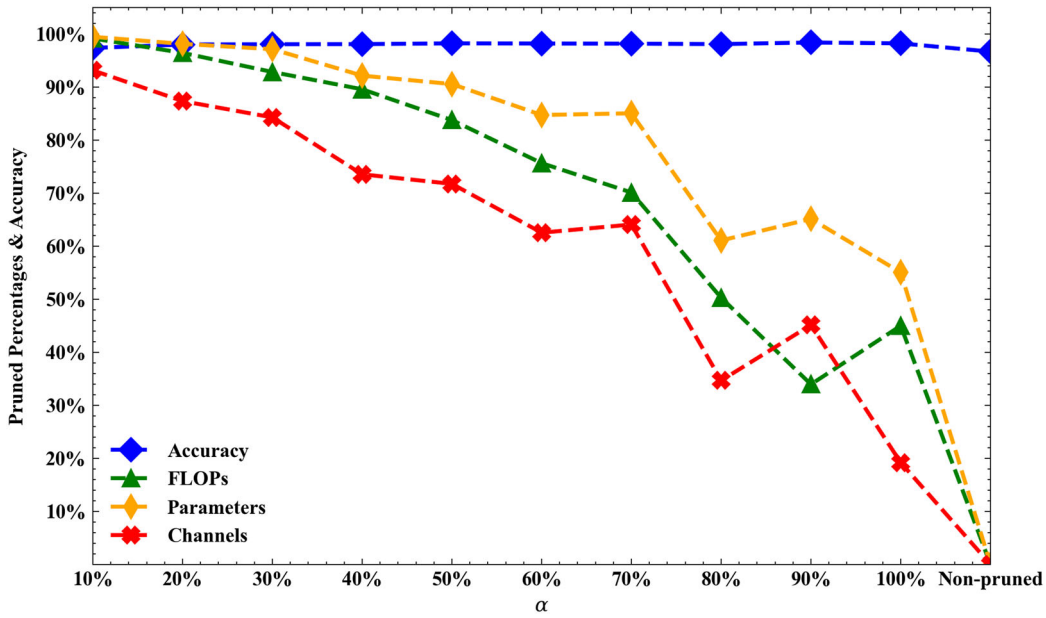


FIGURE 9 The evaluating accuracy and pruned percentages of FLOPs, parameters and channels of the model CNN1D-CACP, which is pruned from CNN1D-CA when α increases from 10% to 100%. "Non-pruned" refers to the model CNN1D-CA, which is not pruned via Algorithm 1. 1D, one-dimensional; CNN, convolutional neural network; CNN1D-CA, 1D-CNN with channel attention; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; FLOP, floating-point operation [Color figure can be viewed at wileyonlinelibrary.com]

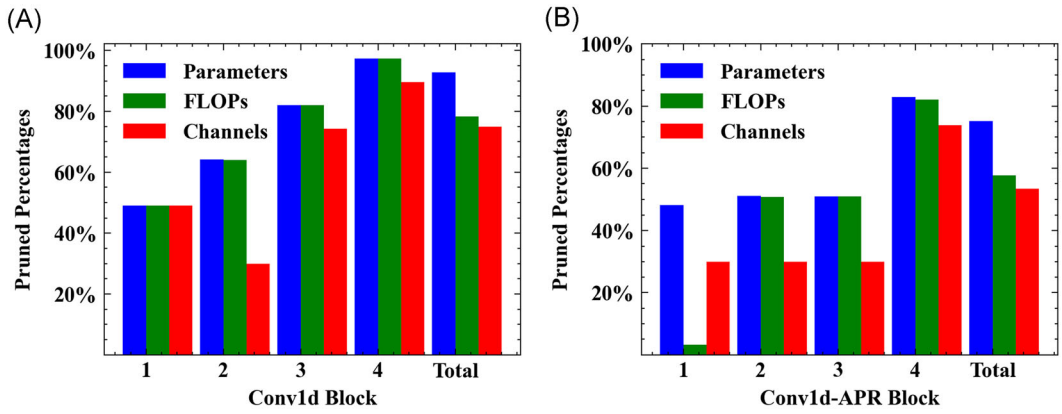


FIGURE 10 The pruned percentages of FLOPs, parameters and channels of CNN1D-CP and CNN1D-CACP in various network depths when $\alpha = 70\%$. "Total" refers to the all blocks of the model. (A) CNN1D-CP. (B) CNN1D-CACP. 1D, one-dimensional; CNN, convolutional neural network; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; CNN1D-CP, 1D-CNN with channel pruning; FLOP, floating-point operation [Color figure can be viewed at wileyonlinelibrary.com]

By combing these two techniques simultaneously (CNN1D-CACP), the recognition rate of three targets, namely decoy 2, decoy 4 and warhead, are all improved. Although these two techniques reduce the recognition rate of decoy 3, it is always higher than 99%.

We can explain the improvements in two aspects. On one hand, the channel attention technique can recalibrate the features extracted by different channels to focus on the information more corresponded to different targets. On the other hand, the channel pruning technique can remove the redundant channels which have obviously negative impacts on extracting discriminative information related to different targets, in other words, it can search

TABLE 4 Average recognition rate (%) of the proposed method against several deep neural networks

Method	Decoy 1	Decoy 2	Decoy 3	Decoy 4	Warhead	Total
ML-ELM	**100.00**	92.87	99.71	96.30	91.34	96.08
SAE-ELM	**100.00**	93.69	99.57	96.06	91.49	96.20
SAE	**100.00**	93.01	96.06	93.93	90.41	94.75
SSpAE	**100.00**	94.86	97.05	95.03	85.05	94.47
SDAE	**100.00**	93.30	96.34	90.90	91.96	94.56
Bi-GRU	**100.00**	94.13	99.57	97.85	86.13	95.45
Bi-LSTM	**100.00**	92.64	99.56	98.25	83.87	94.67
CNN1D	**100.00**	96.49	**99.86**	99.28	88.41	96.69
CNN1D-CA	**100.00**	95.18	99.85	97.90	91.44	96.83
CNN1D-CP	**100.00**	96.63	99.57	98.08	**95.71**	98.06
CNN1D-CACP	**100.00**	**97.03**	99.44	**99.45**	95.62	**98.42**

Note: Absolute best results in each column are boldfaced.

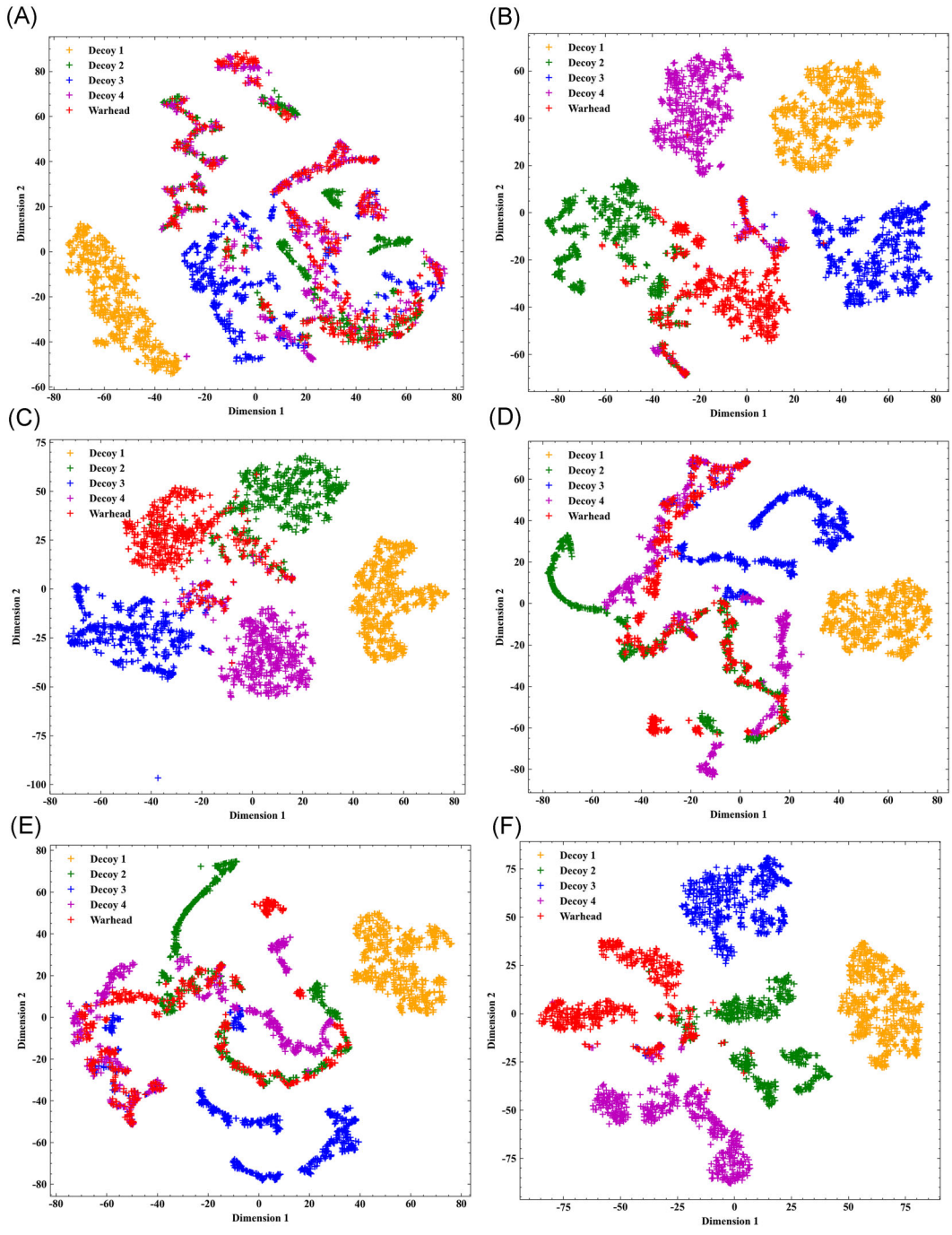
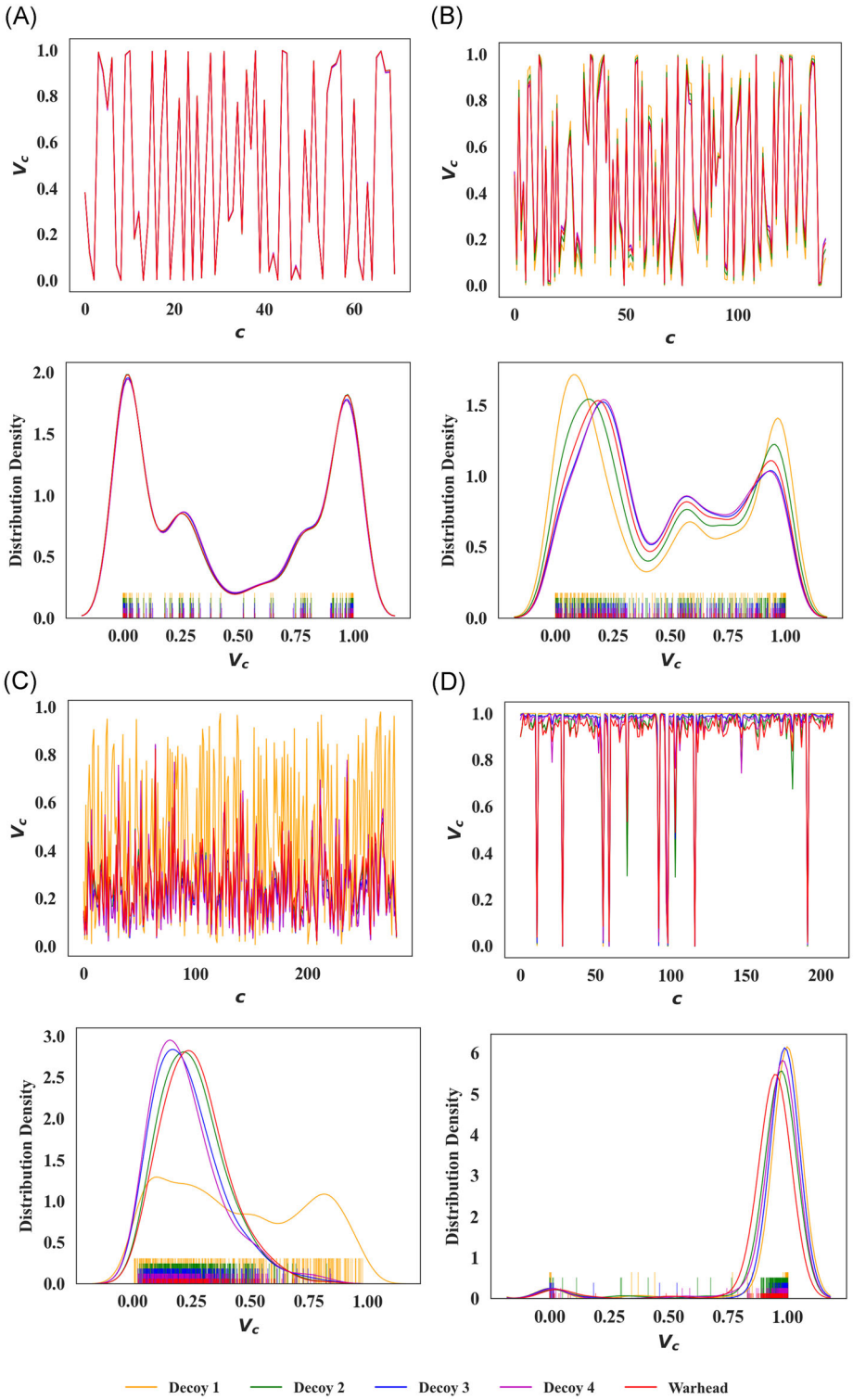


FIGURE 11 Data distribution of all HRRP samples in the test data set after feature extraction by SDAE, SAE-ELM, Bi-GRU, Bi-LSTM and CNN1D-CACP and all extracted features are reduced into two dimensions via t-SNE. (A) measured test data. (B) SDAE. (C) SAE-ELM. (D) Bi-GRU. (E) Bi-LSTM. (F) CNN1D-CACP. Bi-GRU, bidirectional GRU; Bi-LSTM, bidirectional LSTM; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; HRRP, high-resolution range profile; SAE-ELM, stacked auto-encoders and extreme learning machine; SDAE, stacked denoising auto-encoders [Color figure can be viewed at wileyonlinelibrary.com]


(A)
(B)
(C)
(D)
V_c
c
Distribution Density
Decoy 1
Decoy 2
Decoy 3
Decoy 4
Warhead

for a better network structure. Moreover, the two techniques proposed in this paper show behavior of mutually reinforcing between each other. As a whole, they greatly enhance the ability of feature extraction for different channels.

4.5 | Feature visualization

To investigate the discriminative properties of the features extracted by different methods, we utilizethe t-SNE algorithm to map extractedfeatures into two dimensions. Figure 11 compares the separation of the original test data set, SDAE, SAE-ELM, Bi-GRU, Bi-LSTM, and the proposed CNN1D-CACP. As can be seen from Figure 11, targets of decoy 1 always have the discriminative distance to the others in both original data and all compared deep neural networks, deep neural networks chiefly improve the separability of the other four targets. What's more, CNN1D-CACP produces more discriminative feature space than the other methods, since it is capable of searching for better structures of CNNs and enhancing the effective relations between channels.

To provide an assessment of APR blocks in CNN1D-CACP, we study the activations $V_c = \sigma(u_c)$, which is used to recalibrate the features extracted by conv1d blocks in Equation (18). We use all samples in the test data set and examine the distribution of their activations to study the function of this attention mechanism. In Figure 12, we plot the average activations of different targets in various depths and the kernel density estimation (KDE) is adopted to calculate their distribution density. From Figure 12, we can see those activation distributions of different targets in the first APR block are approximately the same, but they are much different in the following blocks. To explain, the earlier layers learn the general features of different targets while later layers learn the features which are more specific to each class. Furthermore, since different samples have different activations, the APR block has the merit of automatically finding the relationships between channels and adaptively enhancing the learned structural features of HRRP.

4.6 | Effect of noise

In this section, we compare CNN1D-CACP with other methods with a different signal-to-noise ratio (SNR) to evaluate the noise robustnessand generalizability,[15] the SNR (dB) is defined as follows:

$$\mathrm{SNR} = 10 \times \log_{10}\left(\frac{\sum_{l=1}^{L} P_l}{L \times P_{\mathrm{Noise}}}\right), \tag{26}$$

FIGURE 12 Average activations of different targets in various CNN1D-CACP network depth and their distribution density calculated by KDE. c refers to the channel index and V_c refers to the average activation in the cth channel. Top lines: the average activations of each target in each channel. Bottom lines: the distribution density of the activations. (A) APR block 1: 70 channels. (B) APR block 2: 140 channels. (C) APR block 3: 280 channels. (D) APR block 4: 209 channels. APR, aggregation-perception-recalibration; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; KDE, kernel density estimation [Color figure can be viewed at wileyonlinelibrary.com]

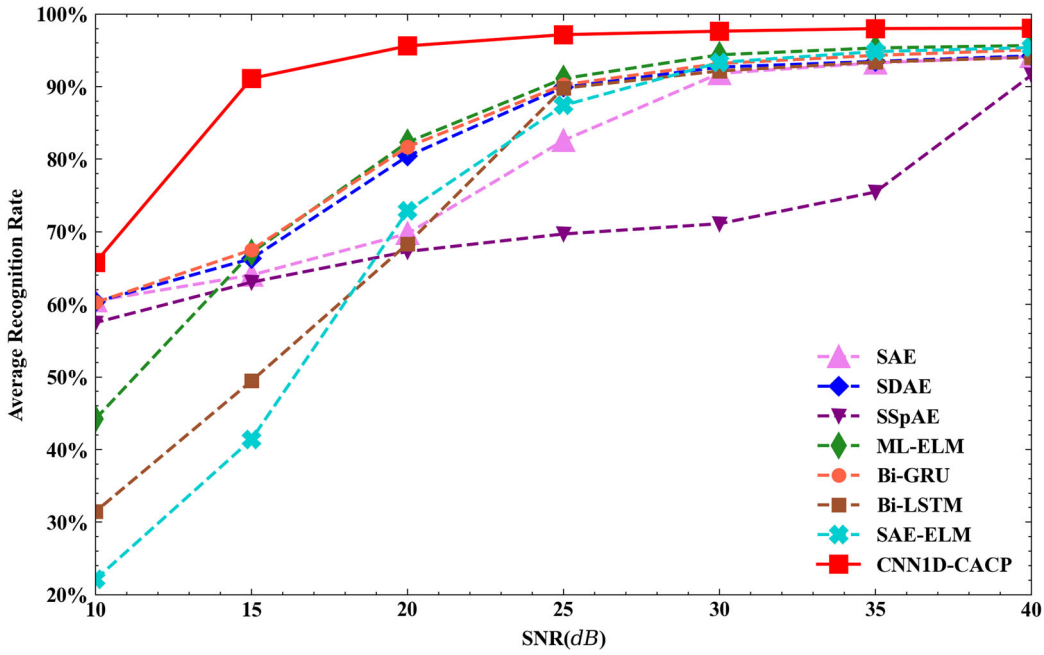


FIGURE 13 The average recognition rate of eight methods versus SNR. Bi-GRU, bidirectional GRU; Bi-LSTM, bidirectional LSTM; CNN1D-CACP, 1D-CNN with channel attention and channel pruning; HRRP, high-resolution range profile; SAE-ELM, stacked auto-encoders and extreme learning machine; SDAE, stacked denoising auto-encoders; SNR, signal-to-noise ratio [Color figure can be viewed at wileyonlinelibrary.com]

where L refers to the number of range cells (here $L = 256$), P_l and P_{Noise} denote the power of the original signal in lth range cell and the power of noise in each range cell, respectively. We add simulated white noise to all the samples in the testing data set with SNR varying from 10 dB to 40 dB. Figure 13 shows the average recognition rate of the methods versus SNR.

As can be seen, CNN1D-CACP achieves consistently higher recognition rates than other methods, proving that the proposed method is more robust to noise in measured HRRP data than these methods. This is because that the proposed CNN-CACP can efficiently extract the structural discriminative features along the range dimensionin HRRP while the fully connected networks cannot. Besides, since all deep neural networks in Figure 13 are trained on the noise-free training data set and tested on the noise testing data sets with different SNR, it also proves that CNN1D-CACP has better generalizability on the noise data sets in some extent.

5 | CONCLUSIONS

This paper presents a novel approach, named CNN1D-CACP, for HRRP-based target recognition. We use a deep neural network interleaved with conv1d, batch normalization and max-pooling layers to extract hierarchy features of HRRP. To enhance the structural information of targets extracted from HRRP, we design an attention block with low computation complexity to recalibrate the learned features adaptively. By introducing the GBL-ABC algorithm for network pruning, the proposed method can automatically find a subnetwork with lower computation complexity and a higher recognition rate than the original network.

Extensive experimental results on measured data demonstrate that the proposed model shows superior performance on classification accuracy, separability in the feature space and robustness to noise than compared deep neural networks. It is noteworthy that although the proposed method has the merits mentioned above, further studies are required to reduce the computational complexity in the pruning phase.

ACKNOWLEDGMENTS

This study was supported by National Natural Science Foundation of China (under grant nos. 61806219, 61876189, 61503407, 61703426, and 61273275), Young Talent Fund of University Association for Science and Technology in Shaanxi, China, under grant no. 20190108, and the Innovation Talent Supporting Project of Shaanxi, China, under grant no. 2020KJXX-065.

REFERENCES

1. Persico AR, Ilioudis CV, Clemente C, Soraghan JJ. Novel classification algorithm for ballistic target based on HRRP frame. *IEEE Trans Aerosp Electron Syst*. 2019;55(6):3168-3189. https://doi.org/10.1109/TAES.2019.2905281
2. Zhao ZC, Wang XD. A minimum risk recognition method of ballistic targets with rejection options. *J Xin Jiaotong Univ*. 2018;52(04):132-138. https://doi.org/10.7652/xjtuxb201804019
3. Zhai XY, Wang XD, Li R, Jia Q. Ballistic target recognition of HRRP based on intuitionistic fuzzy binary tree SVM. *Fire Control Command Control*. 2017;42(10):64-68. https://doi.org/10.3969/j.issn.1002-0640.2017.10.014
4. Li R, Wang XD, Lei L, Xue AJ. Ballistic target HRRP fusion recognition combining multi-class relevance vector machine and DS. *Inf Control*. 2017;46(1):65-71. http://ic.sia.cn/CN/abstract/article_12499.shtml
5. Guo C, Wang HP, Jian T, Xu CG, Sun S Method for denoising and reconstructing radar HRRP using modified sparse auto-encoder. *Chin J Aeronautics*. 2020;33(3):1026-1036. http://d.old.wanfangdata.com.cn/Periodical/hkxb-e202003023.
6. Guo C, He Y, Wang HP, Jian T, Sun S. Radar HRRP target recognition based on deep one-dimensional residual-inception network. *IEEE Access*. 2019;7:9191-9204. https://doi.org/10.1109/ACCESS.2019.2891594
7. Guo C, Jian T, Xu CA, He Y, Sun S. Radar HRRP target recognition based on deep multi-scale 1d convolutional neural network. *J Electron Information Technol*. 2018;41(6):1302-1309. http://jeit.ie.ac.cn//article/id/ed51baaf-2209-4105-bc36-a7bd8713bb69
8. Ding B, Chen P. HRRP feature extraction and recognition method of radar ground target using convolutional neural network. In: *2019 International Conference on Electromagnetics in Advanced Applications (ICEAA)*; 2019:0658-0661. https://doi.org/10.1109/ICEAA.2019.8879042
9. Xiong W, Zhang G, Liu S, Yin J. Multiscale kernel sparse coding-based classifier for HRRP radar target recognition. *IET Radar Sonar Navigation*. 2016;10(9):1594-1602. https://doi.org/10.1049/iet-rsn.2015.0540
10. Xu B, Chen B, Wan JW, Liu HW, Pn L. Target-aware recurrent attentional network for radar HRRP target recognition. *Signal Processing*. 2019;155:268-280. https://doi.org/10.1016/j.sigpro.2018.09.041
11. Wu H, Dai DH, Wang XS. A novel radar HRRP recognition method with accelerated t-distributed stochastic neighbor embedding and density-based clustering. *Sensors*. 2019;19(23):5112. https://doi.org/10.3390/s19235112
12. Liao K, Si JX, Zhu FQ, He XD. Radar HRRP target recognition based on concatenated deep neural networks. *IEEE Access*. 2018;6:29211-29218. https://doi.org/10.1109/ACCESS.2018.2842687
13. Feng B, Chen B, Liu HW. Radar HRRP target recognition with deep networks. *Pattern Recogn*. 2017;61:379-393. https://doi.org/10.1016/j.patcog.2016.08.012
14. Zhou DY. Radar target HRRP recognition based on reconstructive and discriminative dictionary learning. *Signal Process*. 2016;126:52-64. https://doi.org/10.1016/j.sigpro.2015.12.006
15. Du C, Chen B, Xu B, Guo DD, Liu HW. Factorized discriminative conditional variational auto-encoder for radar HRRP target recognition. *Signal Process*. 2019;158:176-189. https://doi.org/10.1016/j.sigpro.2019.01.006
16. Pan M, Jiang J, Kong QP, Shi JG, Sheng QH, Zhou T. Radar HRRP target recognition based on t-SNE segmentation and discriminant deep belief network. *IEEE Geosci Remote Sensing Lett*. 2017;14(9):1609-1613. https://doi.org/10.1109/LGRS.2017.2726098

17. Liu JQ, Chen B, Jie X. Radar high-resolution range profile target recognition based on attention mechanism and bidirectional gated recurrent. *J Radars*. 2019;8(5):589-597. http://radars.ie.ac.cn/cn/article/id/18f357ed-7a0b-479a-a13f-ef04168373f1
18. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1*. Lake Tahoe, Nevada: Curran Associates Inc.; 2012:1097-1105. https://doi.org/10.1145/3065386
19. Zhou FY, Jin LP, Dong J. Review of convolutional neural network. *Chin J Comput*. 2017;40(6):1229-1251. http://www.wanfangdata.com.cn/details/detail.do?_type=perio&id=jsjxb201706001
20. Wan JW, Chen B, Xu B, Liu HW, Jin L. Convolutional neural networks for radar HRRP target recognition and rejection. *Eurasip J Adv Signal Process*. 2019;2019:5. https://doi.org/10.1186/s13634-019-0603-y
21. Wan JW, Chen B, Liu YQ, Yuan YJ, Liu HW, Jin L. Recognizing the HRRP by combining CNN and BiRNN with attention mechanism. *IEEE Access*. 2020;8:20828-20837. https://doi.org/10.1109/ACCESS.2020.2969450
22. Wen Y, Shi LC, Yu X, Huang Y, Ding XH. HRRP target recognition with deep transfer learning. *IEEE Access*. 2020;8:57859-57867. https://doi.org/10.1109/ACCESS.2020.2981730
23. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15(56):1929-1958. http://jmlr.org/papers/v15/srivastava14a.html
24. Goodfellow I, Warde-Farley D, Mirza M, Courville A, Bengio Y. Maxout networks. In: *Proceedings of the 30th International Conference on Machine Learning*; 2013:1319-1327. https://arxiv.org/abs/1302.4389
25. Gohil V, Narayanan SD, Jain A. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers In: *The Thirty-third Conference on Neural Information Processing Systems, NeurIPS*; 2019:2019. https://openreview.net/forum?id=HJeUKEBlLS
26. Frankle J, Carbin M. The lottery ticket hypothesis: finding sparse, trainable neural networks. In: *The Seventh International Conference on Learning Representations, ICLR*; 2019; 2019. https://openreview.net/forum?id=rJl-b3RcF7
27. Liu Z, Sun M, Zhou T, Huang G, Darrell T. Rethinking the value of network pruning. In Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019); 2019: 586-699. https://openreview.net/pdf?id=rJlnB3C5Ym
28. Lin M, Ji R, Zhang Y, Zhang B, Wu Y, Tian Y. Channel pruning via automatic structure search. In: *The Twenty-Ninth International Joint Conference on Artificial Intelligence, JCAI*; 2020;:673-679. https://doi.org/10.24963/ijcai.2020/94
29. Xu F, Li H, Pun C-M, et al. A new global best guided artificial bee colony algorithm with application in robot path planning. *Appl Soft Comput*. 2020;88:106037. http://www.sciencedirect.com/science/article/pii/S1568494619308191
30. Keller JB. Geometrical theory of diffraction. *J Opt Soc Am*. 1962;52(2):116-130. http://www.osapublishing.org/abstract.cfm?URI=josa-52-2-116
31. Wang JJ, Liu Z, Ran L, Xie R. Feature extraction method for DCP HRRP-based radar target recognition via m-$\times$ decomposition and sparsity-preserving discriminant correlation analysis. *IEEE Sens J*. 2020;20(8):4321-4332. https://doi.org/10.1109/JSEN.2019.2962573
32. Zhao CK, He XD, Liang J, Wang TT, Huang CB. Radar HRRP target recognition via semi-supervised multi-task deep network. *IEEE Access*. 2019;7:114788-114794. https://doi.org/10.1109/ACCESS.2019.2933866
33. Wang JJ, Liu Z, Li T, Ran L, Xie R. Radar HRRP target recognition via statistics-based scattering centre set registration. *IET Radar Sonar Navigation*. 2019;13(8):1264-1271. https://doi.org/10.1049/iet-rsn.2018.5625
34. Ma YL, Zhu L, Li YH. HRRP-based target recognition with deep contractive neural network. *J Electromag Waves Applicat*. 2019;33(7):911-928. https://doi.org/10.1080/09205071.2018.1540309
35. Long T, Zhang L, Li Y, Wang YH. Geometrical structure classification of target HRRP scattering centers based on dual polarimetric H/α features. *IEEE Access*. 2019;7:152042. https://doi.org/10.1109/ACCESS.2019.2942425
36. Liu WB, Yuan JW, Zhang G, Shen Q. HRRP target recognition based on kernel joint discriminant analysis. *J Syst Eng Electron*. 2019;30(4):703-708. https://doi.org/10.21629/JSEE.2019.04.08
37. Liu J, Wang BF. Dynamic aircraft identification using HRRP under attitude perturbation interference. *J Electromag Waves Appl*. 2019;33(7):929-945. https://doi.org/10.1080/09205071.2018.1555493

38. Guo PC, Liu Z, Wang JJ. HRRP multi-target recognition in a beam using prior-independent DBSCAN clustering algorithm. *Iet Radar Sonar and Navigation*. 2019;13(8):1366-1372. https://doi.org/10.1049/iet-rsn.2018.5598
39. Lin S, Ji R, Yan C, et al. Towards optimal structured CNN pruning via generative adversarial learning. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2019:2785-2794. https://doi.org/10.1109/CVPR.2019.00290
40. He Y, Lin J, Liu Z, Wang H, Li L-J, Han S. AMC: AutoML for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*; 2018:784-800. https://doi.org/10.1007/978-3-030-01234-2_48
41. Liu Z, Li J, Shen Z, Huang G, Yan S, Zhang C. Learning efficient convolutional networks through network slimming. In: *2017 IEEE International Conference on Computer Vision (ICCV)*; 2017:2755-2763. https://doi.org/10.1109/ICCV.2017.298
42. Wang M, Zhang X-L, Rahardja S. An unsupervised deep learning system for acoustic scene analysis. *Appl Sci*. 2020;10(6):2076. https://doi.org/10.3390/app10062076
43. Liu Q, Furber S. Noisy softplus: a biology inspired activation function. In *International Conference on Neural Information Processing (ICONIP 2016)*; 2016:405-412. https://doi.org/10.1007/978-3-319-46681-1_49
44. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML'15); 2015:448-456. http://proceedings.mlr.press/v37/ioffe15.pdf
45. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; 2018:7132-7141. https://doi.org/10.1109/CVPR.2018.00745
46. 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)*; 2015:1026-1034. https://doi.org/10.1109/ICCV.2015.123
47. Liu L, Jiang H, He P, et al. On the variance of the adaptive learning rate and beyond. In *Eighth International Conference on Learning Representations (ICLR 2020)*; 2020:321-333. https://openreview.net/forum?id=rkgz2aEKDr
48. Zhao XT, Guo K, Sheng XQ. High accuracy scattering center modeling based on PO and PTD. *Radioengineering*. 2018;27:753-761. https://doi.org/10.13164/re.2018.0753
49. Weiyi S, Xiaodan W, Aijun X. Simulation of complete polarized range resolution profile based on FEKO. *J Projectiles Rockets Missiles Guidance*. 2014;34(5):173-175. https://doi.org/10.3969/j.issn.1673-9728.2014.05.043
50. Lekamalage CKL, Song K, Huang G, Cui D, Liang K. Multi layer multi objective extreme learning machine. In: *2017 IEEE International Conference on Image Processing (ICIP)*; 2017:1297-1301. https://doi.org/10.1109/ICIP.2017.8296491
51. Zhao FX, Liu YX, Huo K, Zhang SH, Zhang ZS. Radar HRRP target recognition based on stacked auto-encoder and extreme learning machine. *Sensors*. 2018;18(1):173. https://doi.org/10.3390/s18010173
52. Zhang Z *Research on Range Profile Recognition Based on Deep Learning*. Shandong University; 2019. https://doi.org/10.27272/d.cnki.gshdu.2019.000262
53. Zhao FX, Liu YX, Huo K. Radar target recognition based on stacked denoising sparse autoencoder. *J Radars*. 2017;6(2):149-156. http://radars.ie.ac.cn/article/doi/10.12000/JR16151
54. Xu B, Chen B, Liu J, Wang Q, Liu PH, Radar HW. HRRP target recognition by the bidirectional LSTM model. *J Xidian Univ*. 2019;46(2):29-34. https://doi.org/10.19665/j.issn1001-2400.2019.02.006
55. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. *Adv Neural Inf Process Syst*. 2019;32:8026-8037. http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

How to cite this article: Xiang Q, Wang X, Song Y, Lei L, Li R, Lai J. One-dimensional convolutional neural networks for high-resolution range profile recognition via adaptively feature recalibrating and automatically channel pruning. *Int J Intell Syst*. 2021;36:332-361. https://doi.org/10.1002/int.22302