Sun *et al.*
EURASIP Journal on Audio, Speech, and Music Processing (2025) 2025:22
https://doi.org/10.1186/s13636-025-00409-2

EURASIP Journal on Audio, Speech, and Music Processing

METHODOLOGY

Open Access

ResCapsnet: a capsule network with CRAM and BiGRU for sound event detection

Bing Sun[1], Chenglong Liu[1], Shuguo Yang[1*], Wenwu Wang[2] and Yiduo Mei[3]

Abstract

Sound event detection (SED) is a challenging task where ambient sound events are detected from a given audio signal, which includes categorizing the events and estimating their onset and offset times. Deep learning methods such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have achieved promising performance in SED. However, for overlapping sound events, existing deep learning methods are still limited in detecting individual sound events from their mixtures. Inspired by the success of the dynamic routing mechanism of the capsule network (CapsNet), this paper proposes a capsule network model (ResCapsnet-BiGRU) based on a customized residual attention module (CRAM) and bidirectional gated recurrent unit (BiGRU). CRAM is utilized to extract features from log-mel spectrograms that are relevant to sound events. Through dynamic routing, the capsule network can address the overlapping sound events problem. In addition, the BiGRU with time-distributed fully connected layers is adopted to obtain contextual information. Our proposed method was evaluated on two datasets: the Vehicle Weakly Labeled Sound Dataset (VWLSD , DCASE 2017 Task 4) and the Domestic Environment Sound Dataset (DESD , DCASE 2022 Task 4). It achieved *F*-scores of 62.1% and 75.9% on the Audio Tagging (AT) task, and 54.1% and 59.0% on the sound event detection (SED) task, respectively. The source codes are available at https://github.com/123sunbing/ResCapsnet.git.

Keywords Sound event detection, Capsule network, CRAM, BiGRU, Dynamic routing

1 Introduction

Sound event detection (SED) is an important research direction in the field of audio processing, and its main tasks include localization and classification of sound events. Localization refers to determining the start time and end time of a sound event in the audio stream, i.e., the boundary of the sound event; classification refers to identifying the category of the sound event, such as human voice, car sound, dog barking, and so on. SEDs are widely used in various fields such as machine perception, automatic monitoring, multimedia information retrieval, and anomaly detection [1–3]. Among them, polyphonic sound event detection (PSED) [4] aims to detect sound events from multiple categories, and sound events may occur simultaneously as in Fig. 1. However, existing PSED methods suffer from high error rates. The root cause of this high error rate is the interference and overlap between sound events on the timeline, as evidenced by the challenges of developing practical solutions for PSED [5], and thus it is important to address such challenges for PSED.

Early researchers proposed various statistical modeling approaches for SED, including hidden Markov models (HMM) [6], Nonnegative Matrix Factorization (NMF) [7], Gaussian Mixture Models (GMM) [8], and support vector machines (SVMs) [9], among other statistical modeling approaches. Although these classification methods have good performance, they require complex

*Correspondence:
Shuguo Yang
ysg_2005@163.com
[1] College of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao 266061, China
[2] Centre for Vision Speech and Signal Processing (CVSSP), School of Computer Science and Electronic Engineering, University of Surrey, Guildford GU2 7XH, UK
[3] Inspur Yunzhou Industrial Internet Co., Ltd., Jinan 250101, China

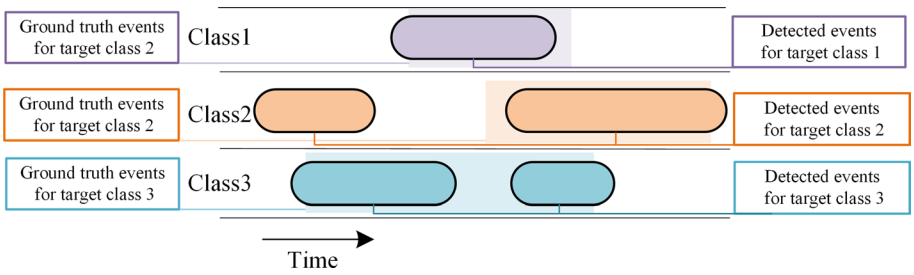

Fig. 1 Three types of detection tasks: clear boxes represent ground truth events, dark squared boxes represent sound event detection

feature extraction methods and are not suitable for large-scale datasets and overlapping sound events [10].

In recent years, with the rapid development of deep neural networks (DNN) and the increasing size of datasets, DNN-based models have become the dominant methods where the audio representations can be automatically learned from data without complex feature extraction methods. Feedforward Neural Networks (FNN) for sound event classification perform significantly better than Support Vector Machines (SVMs) at low signal-to-noise ratio (SNR) levels [11]. Supervised deep learning methods perform well in the absence of prior knowledge and have achieved advanced results in SED tasks [12, 13]. Convolutional Neural Networks (CNNs) were first applied to improve the PSED task in [14–16]. In a recent study, frequency dynamic convolution (FDY conv) [17] used a convolution kernel that varied along the frequency dimension, and the base kernel was a weighted sum using frequency-adaptive attention weights, to obtain a frequency-adaptive convolution kernel. The SED model with FDY conv has achieved state-of-the-art performance on the domestic environment sound event detection (DESED) real validation dataset [17–19].

Capsule networks (CapsNet) [20] have recently been gradually introduced for SED tasks due to their unique capsule structure and dynamic routing mechanism, which separates individual sound events from overlapping mixtures by selecting the most representative spectral features of each sound event [10, 21], giving promising performance for the PSED problem. In another study [22], Iqbal et al. constructed a CapsNet model for the SED task that uses gated convolution in the initial layer and a parallel attention mechanism in the final capsule layer and combines the outputs of these two layers for the final sound event prediction. The algorithm was evaluated on the weakly labeled dataset of the DCASE 2017 challenge [23] and showed great potential. Our work is built on this capsule network architecture as a baseline model.

Although existing capsule network-based frameworks and other DNN-based detection methods have achieved good performance in PSED tasks, simple shallow convolutional networks are difficult to extract high-level features and inadequate in capturing bi-directional dependencies in time-series data and obtaining contextual information. ResNet has shown excellent image classification and object recognition capabilities in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [24]. The technique of "skip connections" is utilized to effectively avoid overfitting of the deep network and the loss of feature information. It also effectively enhances the ability of convolutional blocks for feature extraction in the SED task [25, 26]. Therefore, in this paper, we refine the gated convolution of the baseline system, which consists of gated convolutions with attention layers, and add the technique of "skip connections" of residual networks, which enhances the feature extraction capability of our convolution module, and effectively avoids the problems of gradient vanishing of the deep network. Recurrent Neural Networks (RNNs) [27] are also frequently used in speech recognition [28] and SED [29–31] as a neural network for recognizing patterns in data sequences. The combination of CNNs and RNNs takes advantage of the characteristics of each structure, and it provides state-of-the-art performance, especially in the PSED case [32, 33]. CRNN combines the local shift-invariance of CNNs and the ability to model short-term and long-term temporal dependencies provided by RNN layers. The architecture has also been used in almost all of the best-performing algorithms proposed in several recent research challenges, such as the detection and classification of acoustic scene and event (DCASE) challenge [34]. RNNs perform well in processing sequential data, but they can be limited by the problem of gradient disappearance during back propagation. Long short-term memory (LSTM) [35] and gated recurrent units (GRUs) [36] have been developed to improve RNNs by adding a "gate" structure. LSTM works by introducing three gates (input gate, forget gate, and output gate) and a cell state to control the flow of information. This allows the LSTM to efficiently deliver and retain long-term information and thus better handle long-term dependencies.

GRU is a simplified LSTM that has only two gates (reset and update gates) and but no cell state, instead it directly takes the hidden state as an output. In this paper, we choose Bidirectional GRU (BiGRU) [37] as the module to process time series data after the capsule layer. This model captures the fine-grained correlation between the data by accessing the two sequence directions, and efficiently acquires the contextual information. It involves fewer parameters and less computation than LSTM.

In this paper, we propose a new model architecture Res-Capsnet-BiGRU for PSED. The model mainly consists of three modules, CRAM, capsule layer, and BiGRU Model. Specifically, we use CRAM instead of the traditional convolutional layer to extract high-level features that closely correlate with sound events, while ignoring irrelevant information such as background noise. Then, a dynamic routing mechanism is introduced in the capsule layer to effectively detect overlapping sound events. To obtain contextual information, BiGRU with time-distributed fully connected layers is adopted after the capsule layer. We conducted experiments on the VWLSD [38] and the DESD [39, 40]. The experimental results show that, compared with the baseline system [22], the method proposed in this paper has significant performance improvements.

- We introduce a customized residual attention module (CRAM) to improve the audio features extracted from the input log-mel spectrogram.
- We introduce the BiGRU model to better capture the long-term dependent information in sequence data as well as the temporal context information.
- We perform extensive experiments to show the improved performance of our proposed model.

2 Capsule layer

The model of capsule networks was originally proposed in [20]. The main idea is to transform the inputs and outputs of capsule neurons from scalar to vector form to reduce the loss of feature information and improve the SED ability of the model.

The capsule network mainly consists of PrimaryCaps and DigitCaps, where PrimaryCaps, also known as the low-level capsule layer, includes convolution, reshaping and compression, and uses Relu as a nonlinear activation function. And DigitCaps is the high-level capsule layer of the capsule network, which is responsible for integrating the low-level features (vectors output from the PrimaryCaps layer) into high-level semantic information through the dynamic routing mechanism, and finally outputs the classification results. It is the core of the whole model and makes appropriate prediction of the final classification by learning the positional relationship between the local and the whole. Simply speaking, the low-level capsules are used to detect the probability of occurrence and gesture of some specific patterns, and the high-level capsules are used to detect more complex dynamic patterns. Each capsule consists of several neurons, and the output of each neuron represents a different property of the same object. This provides a great advantage in object recognition, i.e., recognizing the whole by identifying some of the properties of an object. For example, when multiple sound events overlap in the time or frequency domain (e.g., dog barking and human voice), traditional CNNs may confuse different sound event features due to the local receptive fields of the convolutional operation, whereas capsule networks construct the spatial-temporal distribution structure of the frequency and duration of sound events through the "gesture matrix", which assigns features with different frequencies at the same point in time (e.g., a high-frequency bird chirping and a low-frequency engine sound) to different capsules, and at the same time, associates features with the same frequency at different time segments (e.g., intermittent knocking on the door) with the same frequency. At the same time, features of the same frequency (e.g., intermittent knocking) are associated with different time segments.

Table 1 lists the differences between capsule vector neurons and traditional scalar neurons, where x_i, $i = 1, 2, ..., n$, represent the inputs to the traditional neurons, u_i, $i = 1, 2, ..., n$, represent the low-level capsules. $\hat{u}_j$, $j = 1, 2, ..., n$ represent the predictions from u_i to the high-level capsule v. w_i, $i = 1, 2, ..., n$, represent the corresponding weights, b represents the bias, and $\sum$ represents the weighted sum. $f(x)$ and $Squash(x)$ stand for the activation functions.

The dynamic routing algorithm is introduced to match the capsule representing the timeframe in PrimaryCaps with the capsule representing the event characteristics in DigitCaps. Its dynamic routing process is shown in Fig. 2, and a nonlinear squash function is used to normalize the capsule output within the range of [0,1] as shown in Eq. (1).

$$V_j = \left\{ \|S_j\| / \left(1 + \|S_j\|^2\right) \right\} \cdot \left(S_j / \|S_j\|\right) \tag{1}$$

Table 1 Differences between capsule vector neurons and traditional neurons

Capsule vector neurons vs. traditional neurons		
Input	u_i	x_i
Affine transformation	$\hat{u}_{j\|i} = W_{ij} u_i$	-
Weighted sum	$s_j = \sum c_{ij} \hat{u}_{j\|i}$	$a_j = \sum_i w_i x_i + b$
Nonlinear activation	$v_j = \left\{ \|S_j\| / \left(1 + \|S_j\|^2\right) \right\} \cdot \left(S_j / \|S_j\|\right)$	$h_j = f(a_j)$
Output	v_j	h_j

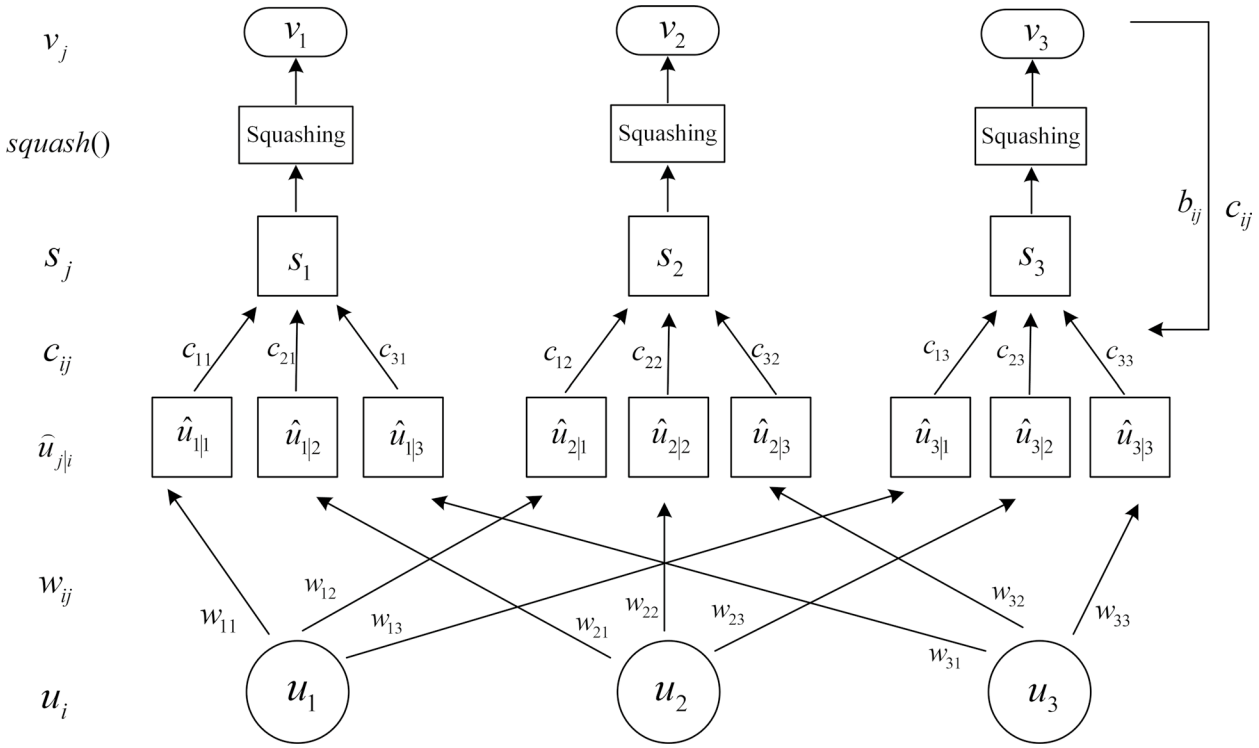

Fig. 2 Dynamic routing

here, v_j is the output vector of high-level capsule j and s_j is its total input.

$$s_j = \sum c_{ij} \cdot \hat{u}_{j|i} \tag{2}$$

where s_j is the weighted sum of all prediction vectors $\hat{u}_{j|i}$ generated by low-level capsule i and passed to high-level capsule j,

$$\hat{u}_{j|i} = W_{ij} u_i \tag{3}$$

while c_{ij} is computed as Softmax b_{ij}, which is the logarithmic prior probability between the low-level capsule i and high-level capsule j

$$c_{ij} = exp(b_{ij}) / \sum_k (b_{ik}) \tag{4}$$

where c_{ij} is iteratively refined in terms of b_{ij}, with b_{ij} initialized to 0. The consistency between v_j and $\hat{u}_{j|i}$ is measured, and the higher the similarity between $\hat{u}_{j|i}$ and v_j, the larger the increment of c_{ij}. This consistency is given by the scalar product $a_{ij} = v_j \cdot \hat{u}_{j|i}$. The coefficients b_{ij} are updated as follows:

$$b_{ij} \longleftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j \tag{5}$$

The whole process reflects the dynamic routing of all low-level capsule i in the L layer and all high-level capsule j in the $L+1$ layer. Due to this attribute of dynamic routing, CapsNet is not only able to handle rich features and enable multi-level interactions between neurons but also allows for full training of the model without the need for extensive data augmentation or a proprietary domain adaptation process, which provides a significant boost to PSED performance [21].

3 Proposed method

3.1 Overview

In this section, we present a new model, i.e., the Res-Capsnet-BiGRU model, for sound event detection, which consists of three modules, namely, the customized residual attention module (CRAM), capsule layer, and the BiGRU module. To extract high-level features of audio, we propose the CRAM, where we introduce two series-connected attention modules, which consists of gated convolutions with an attention layer to capture audio channel information. This is followed by a capsule layer with dynamic routing which facilitates the detection of overlapping sound events. Further, we introduce the BiGRU model to capture the contextual information and the bidirectional dependencies of the audio time series data. The detailed architecture of the model proposed is shown in Fig. 3.

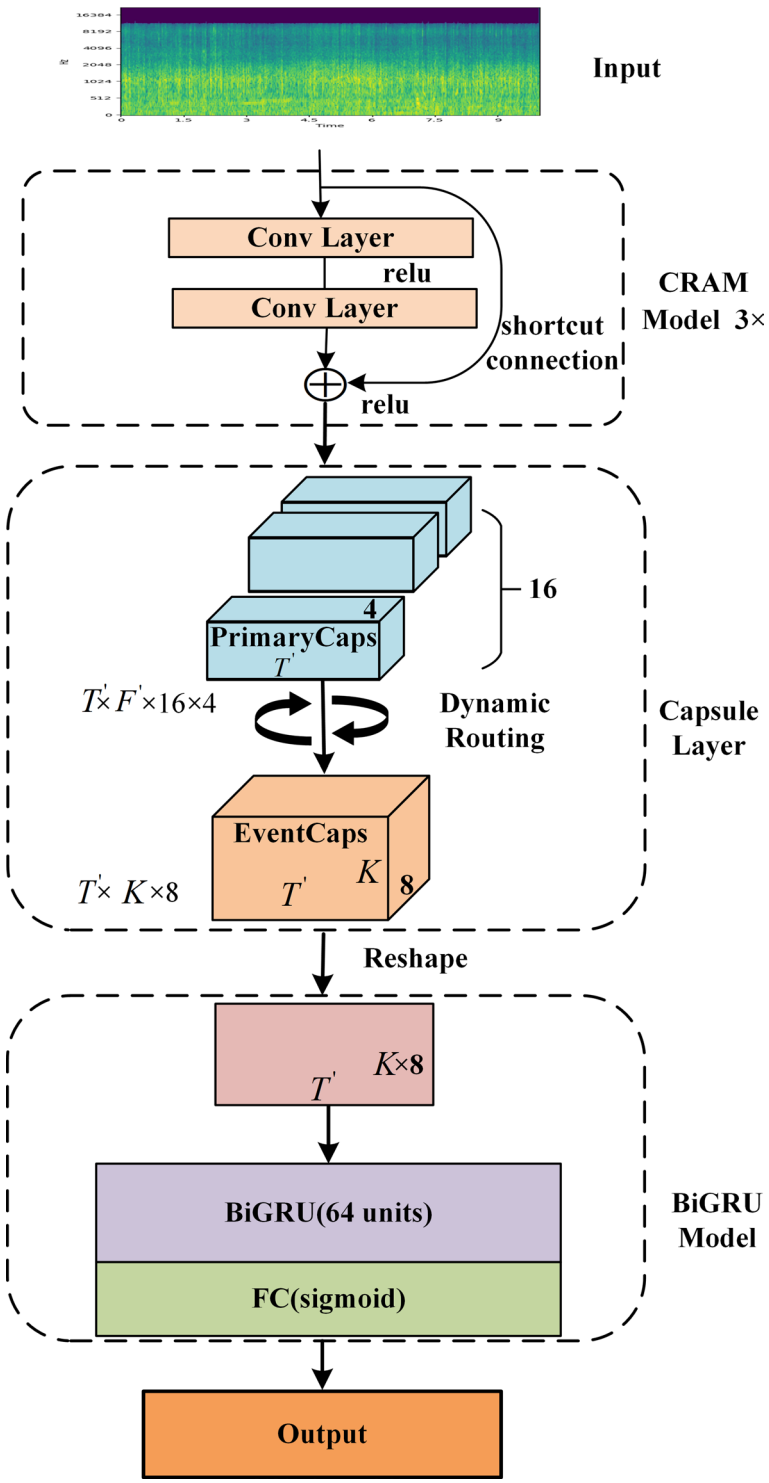

Fig. 3 The architecture of ResCapsnet-BiGRU model

1) In the CRAM, the log-mel spectrogram is used as input to extract relevant audio features and convert the time-varying audio signal into a feature vector suitable for subsequent detection.

2) In the capsule layer, PrimaryCaps and DigitCaps are connected through a so-called dynamic routing that encourages learning part-whole relationships and improves the detection performance of the model in PSED.

3) In the BiGRU module, the time-distributed fully connected layer is attached to learn the temporal context information produced by the capsule layer with attention features and to estimate the probability of event activities.

3.2 Customized residual attention module (CRAM)

We concatenate three CRAM blocks sequentially to extract features from the input audio representation, as shown in Fig. 4. In each CRAM, we introduce two attention modules, each consisting of a gated convolution with the attention layer, formulated as follows:

$$Y = \varphi(f(X) \odot \sigma(f(X))) \tag{6}$$

$$Z = \varphi(f(Y) \odot \sigma(f(Y))) \tag{7}$$

where X is the input feature map, which is a log-mel spectral feature map of dimension $240 \times 64 \times 1$; $f(\cdot)$ denotes a gated convolution operation with 64-channel kernel size 3×3 and stride 1; σ is the *sigmoid* activation function for generating the attention weights. $\odot$ is the Hadamard product operator and φ is the *Relu* activation function. To improve the extracted features, the output Y of the first attention module is further applied with one more attention module, as shown in Eq. (7).

In addition, to prevent the gradient vanishing problem, we introduce "skip connections" in the residual network, which allow us to bypass the attention module and directly merge the input X and output Z of the module to obtain the feature map $X^{'}$, as shown in Fig. 4. The process is formulated as follows:

$$X^{'} = X + Z \tag{8}$$

where the shapes of the feature maps X, Z and $X^{'}$ are denoted as $h \times w \times c$, in terms of their heights, width and number of channels. In our experiments, for the first CRAM, the log-mel spectrogram has a shape $240 \times 64 \times 1$, while the feature maps Z and $X^{'}$ both have a shape $240 \times 64 \times 64$.

Finally, we used a max-pooling layer to halve the dimension of the frequency axis of the CRAM output feature map $X^{'}$, but maintaining the dimension of time axis, and followed by a nonlinear activation function *Relu* .

3.3 The CapsNets layer

Following the baseline system [22], we also use the capsule network module for the detection of overlapping audio events and capturing the spatial relationships among the events. The capsule network is composed of PrimaryCaps and DigitCaps.

The first layer of PrimaryCaps, also called low-level capsules, consists of convolution, reshaping, and squashing, and uses *Relu* as a nonlinear activation function. First, the convolution acts on the feature map $X^{'}$ obtained after three CRAM convolutions, and after convolution and activation, PrimaryCap generates a low-level feature vector u_i, $i = 1, 2, ..., n$, of dimension $T^{'} \times F^{'} \times 16 \times 4$, where $T^{'}$ and $F^{'}$ are the number of frames and frequency bins obtained after convolution activation, and 16 is the number of PrimaryCaps channels and each channel consists of a 4-dimensional capsule. Here, 64 filters are used with a kernel size of 3, and the time and frequency dimensions are set to 1 and 2, respectively. After that, it is input to DigitCaps after the reshaping and batch normalization operations, which we also call high-level capsules. Since the previous layer is also a capsule layer, a dynamic routing algorithm is used between the two capsule layers, which matches the capsule representing the time frame in PrimaryCaps with the capsule representing the event features in DigitCaps, as described in detail in Sect. 2.

The output of the low-level capsule u_i, $i = 1, 2, ..., n$, is first multiplied by the weight matrix w_i, $i = 1, 2, ..., n$, to compute the prediction vector $\hat{u}_j$, $j = 1, 2, ..., n$ of the high-level capsule. Then they are weighted sum to obtain the output vector v, whose weight is determined by the dynamic routing, which aims at allowing the bottom-level capsule to autonomously choose the optimal path for the information to be propagated to the high-level capsule. Finally, the output vector is nonlinearly mapped to obtain a high-level feature vector of dimension $T^{'} \times K \times 8$, where 8 is the dimension of DigitCaps, and K is the number of sound event classes.

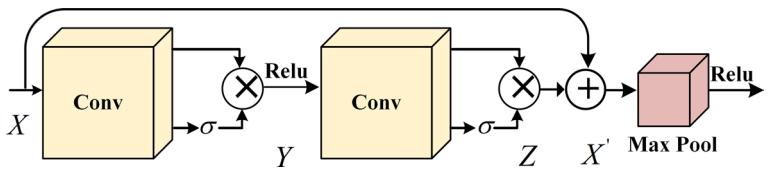

Fig. 4 Customized residual attention module

3.4 BiGRU model

Temporal dependencies are important in sound event analysis tasks as shown in [29, 41]. To capture the bi-directional dependencies, we introduce the BiGRU module for representing contextual information. This module consists of BiGRU and a fully connected layer.

The BiGRU module consists of two independent gated recurrent units (GRUs), as shown in Fig. 5. The output of the capsule layer is reshaped into a two-dimensional tensor $T^{'} \times (K \times 8)$, which is later inputted into the BiGRU, as follows:

$$\overrightarrow{h_t} = GRU(\overrightarrow{h_{t-1}}, x_t) \tag{9}$$

$$\overleftarrow{h_t} = GRU(\overleftarrow{h_{t+1}}, x_t) \tag{10}$$

$$y_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{11}$$

First, the input sequence is a tensor of $X = [x_0, x_1, ..., x_t]$, with x_t denoting the t-th element in the input sequence. The forward GRU is responsible for processing the time series information from the beginning of the sequence to the current time step to produce a hidden forward sequence $\overrightarrow{h_t}$, while the reverse GRU is responsible for processing the time series information from the current time step to the end of the sequence to compute a reverse hidden sequence $\overleftarrow{h_t}$. These two GRU units are run in parallel, and the outputs are merged to form the final output y_t. The symbol $[.; .]$ denotes the vector splicing operation.

To obtain the probability of event activity per frame, we pass the hidden state output y_t to a feedforward layer with a sigmoid activation function to obtain a tensor of shape $T^{'} \times K$, with K being the number of sound event classes.

The output of the BiGRU Model is a frame-level prediction, i.e., the probability of sound event activity on each frame. However, in this paper, we are using weakly labeled data where only clip-level labels are available. Therefore, we need to aggregate the frame-level predictions into the clip-level predictions by the softmax function defined as follows,

$$y_l = \sum_i y_i exp(y_i) / \sum_i exp(y_i) \tag{12}$$

where $y_i \in [0, 1]$ represents the frame-level predicted probability of a class of sound events and $y_l \in [0, 1]$ represents the clip-level aggregation probability of the events. We then set a threshold τ_1 to detect the presence of event l. When y_l τ_1, sound event l exists. To calculate time information, we threshold the probabilities of y_i with another value τ_2, then the onset and offset times are determined from the obtained binary matrix.

4 Experiments

4.1 Dataset and performance metrics

Our study utilized the Vehicle Weakly Labeled Sound Dataset (VWLSD) from DCASE 2017 Task 4 and the Domestic Environment Sound Dataset (DESD) from DCASE 2022 Task 4 for experiments. Both datasets are subsets of AudioSet, with each audio clip lasting 10 s and corresponding to one or multiple sound events. VWLSD consists of 17 sound event categories, including 9 warning sounds and 8 vehicle sounds. It is divided into a training set (51,172 audio clips), a validation set (488 audio clips), and an evaluation set (1103 audio clips). The DESD contains 10 sound event categories from daily life scenarios, such as *Dishes* and *Speech*. It includes a training set (4784 real audio clips and 10,000 synthesized audio clips), a validation set (1000 audio clips), and a test set (360 audio clips). In these datasets, two tasks are performed for evaluation: the AT and the SED, where the AT aims at predicting the type of sound events contained in the

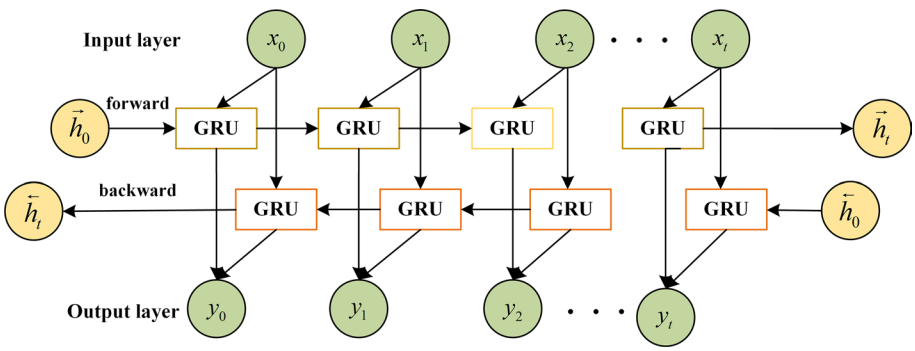

Fig. 5 BiGRU architecture

audio clips, and the SED detects the onset and offset of each sound event, apart from predicting its class.

For these two tasks, our study uses the evaluation metrics including Precision (P), Recall (R), and *F*-score (F1), which can measure the performance of SED, as in [42]. In this paper, F1 is used to evaluate the performance of SED.

4.2 Experimental setup

All experiments are performed by the TensorFlow framework. GCCaps [22] is chosen as the baseline model. The key parameters and structures of the proposed model are shown in Tables 2 and 3. More specifically, Table 2 details the parameters of CRAM in the feature extraction process, while Table 3 details the parameter settings of PrimaryCaps, DigitCaps, and BiGRU modules.

The audio signal sampled at 16,000 Hz is converted to log-mel spectrogram as input to the system. The log-mel spectrograms were computed using a 64 ms frame length, 20 ms overlap between neighboring windows, and 64 mel-frequency bins per frame. For each 10 s sample, this gives a feature vector of dimension 24×64.

To speed up convergence and prevent overfitting, batch normalization [43] and dropout [44, 45] operations are used after the convolutional layer, and batch normalization operation is used after the initial capsule layer. The dropout rate for the convolution in CRAM was set to 0.2, and in the initial capsule layer to 0.5.

In the training stage, the batch size was set to 44, and the initial learning rate was set to 0.001. The network was trained for a total of 30 epochs, with every two epochs decaying by a factor of 0.9, and in each epoch, the learned weights were saved. The number of dynamic routing iterations was set to $r = 3$. Binary cross-entropy was used as the loss function and Adam [46] as the optimizer.

The trainable parameters of our model are 486,810, in which the number of parameters of the 3-layer main CRAM module is about 410,500 and FLOPs is about 18 B. The parameters of BiGRU and full connection layer are 55,680 and 20,480, respectively, and FLOPs are 2.24 M and 1.23 M, respectively.

For VWLSD, the number of events in the test and evaluation sets is balanced, while the number of events in the training set is not balanced, which may lead to classification bias. The data balancing technique proposed in [47] is used to minimize the impact of this problem.

During inference, the models that achieved the highest accuracy on the validation set were selected and their predictions were averaged. The detection thresholds were set to $\tau_1 = -1$ and $\tau_2 = 0.4$ for our system. For SED, the dilation and erosion sizes were set to 10 and 5, respectively. For the other hyper-parameters, these values were determined based on experiments on the validation set.

Table 2 Model parameters (feature extraction)

	Feature extraction		
	Conv1	**Conv2**	**Conv3**
Kernel size	64@3×3	64@3×3	64@3×3
Stride	1×1	1×1	1×1
Pooling size	2×2	2×2	2×2
Dropout rate	0.2	0.2	0.2
Activation function	Relu	Relu	Relu

Table 3 Model parameters (capsule layers, BiGRU model)

	Capsule layers		BiGRU Model	
	Primary Capsule	**Event Capsule**	**GRU**	**FC**
Kernel size	32@3×3	–	–	–
Step	1×2	–	–	–
Activation function	Squashing	Squashing	Tanh	Sigmoid
Number of hidden units	–	–	64	17
Capsule dimension	4	8	–	–

4.3 Results and discussion

The experiments are performed in four aspects, i.e., choice of thresholds, CRAM for feature extraction, comparison between the proposed ResCapsnet-BiGRU and other methods, and ablation experiments.

4.3.1 Empirical choice of thresholds

In order to find a suitable threshold, we used a baseline system to conduct experiments on the VWLSD by adjusting the thresholds, i.e., first setting the AT threshold $\tau_1 = -1$, and a more appropriate threshold was selected by adjusting the threshold τ_2 of the SED using the 0.1 step size setting. The related results are shown in Table 4.

With a higher threshold, although false alarms are reduced, some events are mis-detected, leading to higher

Table 4 Performance of the baseline method with various thresholds

Thresholds	**AT results**		**SED results**	
	***F*-score**	**EER**	***F*-score**	**EER**
0.1	58.4%	15.6%	35.9%	2.46
0.2	58.6%	14.5%	44.1%	1.31
0.3	57.8%	15.1%	45.7%	0.8
0.4	58.9%	14.4%	45.5%	0.8
0.5	57.9%	15.6%	40.8%	0.79
0.6	57.4%	15.8%	36.5%	0.81

Table 5 By the comparison between not using CRAM and using CRAM for feature extraction

Evaluation metrics	Not using CRAM		Using CRAM	
	AT results	SED results	AT results	SED results
F-score	58.6%	46.3%	59.4%	46.7%
Precision	59.2%	58.3%	54.6%	52.2%
Recall	57.9%	38.4%	65.2%	42.2%
EER	-	0.76	-	0.82

precision and lower recall. Vice versa, for a lower threshold, more false alarms might be introduced. We found empirically that a higher *F*-score is achieved for both AT and SED using a threshold 0.4. Therefore, subsequent experiments will be based on the thresholds $\tau_1 = -1$ and $\tau_2 = 0.4$.

4.3.2 CRAM for feature extraction

Using the thresholds identified in Sect. 4.3.1, we perform experiments to study the impact of using CRAM for feature extraction. The results are shown in Table 5.

From the results, we can find that after the introduction of the CRAM, the *F*-score for AT and SED achieves a significant improvement. Specifically, on the VWLSD, the *F*-score of AT is increased from 58.6 to 59.4% and that of SED from 46.3 to 46.7%. This result suggests that using CRAM can further validate its advantages in feature extraction, thus improving the SED performance of the model.

4.3.3 Comparison with other methods

The proposed method is compared with GCCaps [22], GCRNN [47], GCNN [22], and CRNN [48]. GCCaps is the baseline system on which the proposed method is built. GCRNN uses gated convolutional and recurrent layers [49] instead of capsule layers. It achieved first place in the audio labeling subtask of Task 4. GCNN [22] model is similar to GCRNN but not including the recurrent layers. The results for AT and SED are shown in Tables 6 and 7, respectively. On the VWLSD, our method outperformed the comparable models on both AT and SED tasks, achieving the *F*-scores for AT and SED at 62.1% and 54.1%, respectively. By using CRAM for feature extraction, deeper channel features are successfully extracted, which significantly improves the performance of audio detection. In the SED task, the recurrent layer significantly improves the localization of ResCapsnet-BiGRU and GCRNN because its scores are much higher than that of GCNN, and Recall is also much better. Compared to GCRNN, our ResCapsnet-BiGRU achieves a 10.8% improvement in *F*-score. This indicates that the combination of the capsule layer and BiGRU in our method enhances the recurrent layer in the GCRNN, and improves the experimental performance of the PSED. In addition, compared to the Capsule-transformer [50] proposed in 2024, our strategy improves by 1.5% and 6.2% on AT and SED tasks, respectively.

On the DESD, ResCapsnet-BiGRU achieves better performance, with 77.3% and 75.9% F in AT and SED tasks, respectively. Mainly due to the obvious heterogeneity

Table 6 Performance results of audio tagging subtask

Method	DataSet	*F*-score	Precision	Recall
GCNN	VWLSD	57.2%	59.0%	57.2%
GCRNN	VWLSD	57.3%	53.6%	59.6%
GCCaps	VWLSD	58.6%	59.2%	57.9%
Capsule-transformer	VWLSD	60.6%	62.9%	57.6%
ResCapsnet-BiGRU	VWLSD	62.1% (61.7%–62.5%)	57.4% (57.0%–57.8%)	67.7% (67.3%–68.1%)
ResCapsnet-BiGRU	DESD	77.3% (76.4%–78.2%)	77.5% (76.6%–78.4%)	77.2% (76.1%–78.3%)

Table 7 Performance results of sound event detection subtask

Method	DataSet	*F*-score	Precision	Recall
GCNN	VWLSD	37.5%	46.6%	31.1%
GCRNN	VWLSD	43.3%	57.9%	34.8%
GCCaps	VWLSD	46.3%	58.3%	38.4%
Capsule-transformer	VWLSD	47.9%	68.7%	29.1%
ResCapsnet-BiGRU	VWLSD	54.1% (53.7–54.5%)	48.4% (48.0–48.8%)	61.3% (59.8–61.8%)
ResCapsnet-BiGRU	DESD	75.9% (75.4–76.4%)	75.0% (74.6–75.4%)	76.8% (76.4–77.3%)

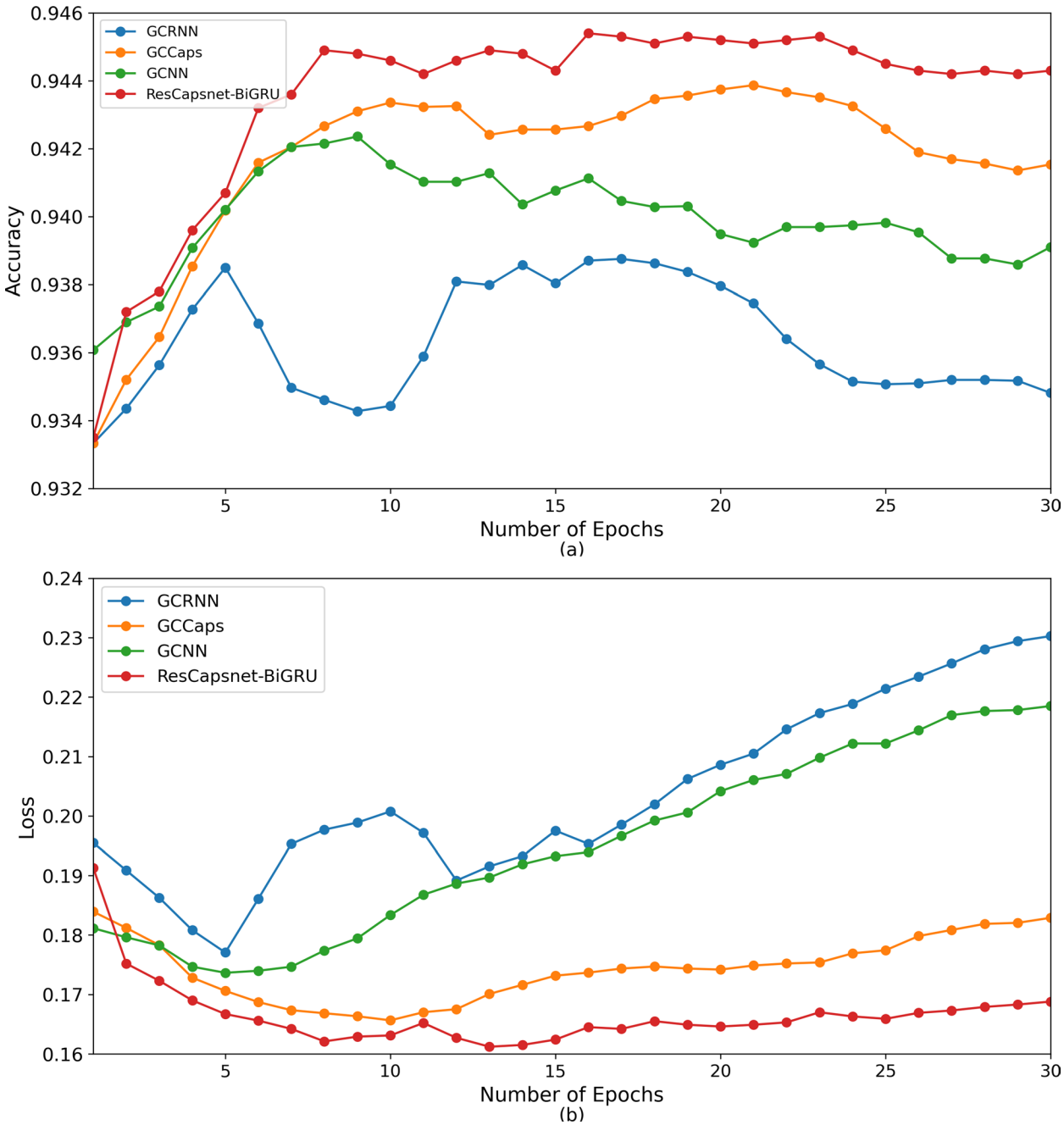

Fig. 6 Performance as a function of the number of epochs for (**a**) accuracy and (**b**) loss. The proposed ResCapsnet-BiGRU model has the highest accuracy and the lowest loss

Table 8 F-score of audio tagging subtask for each event

Train horn	Air horn, truck horn	Car alarm	Reversing beeps	Bicycle	Skateboard	Ambulance	Fire engine, fire truck	Civil defense siren
81.5%	64.7%	65.5%	67.9%	49.4%	72.7%	58.6%	67.3%	87.9%
Police car	Screaming	Car	Car passing by	Bus	Truck	Motorcycle	Train	Micro average
65.3%	88.5%	66.3%	38.2%	43.1%	48.9%	60.3%	79.2%	62.1%

Table 9 Error rate of sound event detection subtask for each event

Train horn	Air horn, truck horn	Car alarm	Reversing beeps	Bicycle	Skateboard	Ambulance	Fire engine, fire truck	Civil defense siren
0.67	0.85	0.72	0.80	1.7	0.88	0.94	0.86	0.33
Police car	Screaming	Car	Car passing by	Bus	Truck	Motorcycle	Train	Micro average
0.85	0.7	0.84	1.4	1.46	1.32	0.88	0.73	0.79

between the 10 sound event classes in DESD (e.g., significant differences in temporal dynamics and spectral patterns between speech and dishes) that contribute to discriminative feature learning. This inherent diversity enables models to develop more robust class-specific representations, ultimately leading to improved classification performance.

To obtain greater insight, we also compared the performance of these models on the validation set as a function of the number of epochs. As evident in Fig. 6, our proposal achieved the highest accuracy and lowest loss. In Fig. 6a, it can be seen that the accuracy decreases after a number of epochs. These issues are not observed with the training set, which suggests that the models are overfitting. However, as shown in the figures, the extent of this problem is greatly reduced when the network models GCCaps and ResCapsnet-BiGRU for capsule routing are used. It can be seen in Fig. 6b that all of the models eventually diverge in terms of the value of the loss function.

Tables 8 and 9 show the model results for all sound events on the two subtasks on the VWLSD. In the AT subtask, "Civil defense siren, Train horn" and "Screaming" events were classified with higher accuracy, and "Car passing" and "Bus" events were classified with lower accuracy. In the SED subtask, events such as "Civil defense siren" had low error rates, while events such as "Bicycle", "Car Passing by", and "Bus" had higher error rates.

These results show that the use of improved capsule dynamic routing leads to improved accuracy in the AT subtask and reduced error rates in the SED subtask. Specifically, the introduction of the CRAM can fully utilize the raw feature information, allowing the model to learn more audio features. The routing algorithm of the capsule network enables the model to identify the whole-local relationship, which enhances the generalization ability of the model. The introduction of BiGRU model enables the model to learn contextual information, which enhances the temporal localization ability of the model. Therefore, the model achieves satisfactory performance on both the AT subtask and the SED subtask.

4.3.4 Ablation experiments

Change the number of hidden units in the BiGRU model. Figure 7 shows the experiments under varying the number of hidden units in the BiGRU Model, and we can see that both AT and SED achieve a significant improvement in *F*-score compared to the baseline system. In particular, the best model performance is achieved when the number of hidden units is set to 64. The *F*-score of AT improves to 62.1%, which is a 3.5% increase compared to the baseline, and the *F*-score of SED improves to 54.1%, which is a 7.8% increase compared to the baseline. These results show that for our model, with the number of hidden units of the BiGRU Model set to 64, the model is better able to bi-directionally process the time sequence data and learn the rich temporal information, which results in a significant enhancement in the localization and detection of sound events.

Changing the model learning rate. Figure 8 shows the experiment under varying model learning rate, in which case our model decreased the *F*-score results with the increase in learning rate and increased the ER

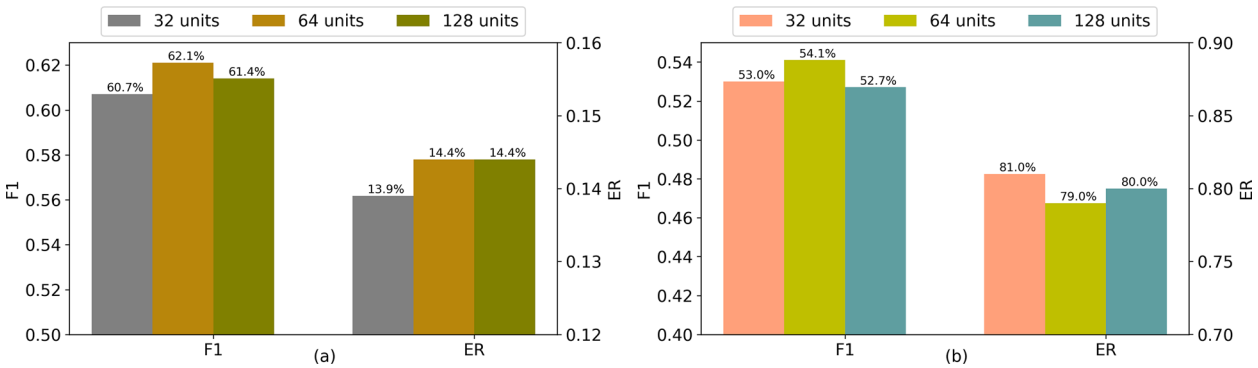

Fig. 7 **a** and **b** represent the AT results and SED results of the model with different numbers of hidden units in BiGRU, respectively

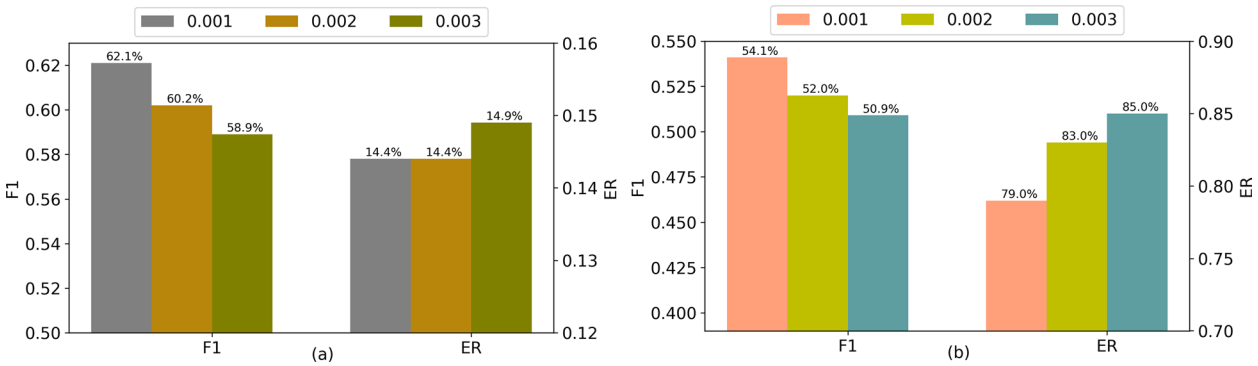

Fig. 8 **a** and **b** represent the AT results and SED results, respectively, for changing the model learning rate

results instead. The *F*-score of AT is decreased from 62.1 to 58.9% and the *F*-score of SED is decreased from 54.1 to 50.9% when the Learning rate is increased from 0.001 to 0.003. This result shows that when the learning rate is too large, it can lead to unstable model training and the model parameters are updated too much during the training process, which leads to the possibility of the model ignoring more audio time information during the learning process. Based on the experiments, the learning rate of 0.001 was determined to be suitable for our model, which not only effectively accelerates the training of the model, but also improves the accuracy of the model.

5 Conclusion

In this study, a hybrid model of CRAM and BiGRU on the basis of capsule architecture has been presented for the problem of polyphonic sound event detection. This model utilizes the powerful feature extraction capability of CRAM to extract richer and more relevant audio features for SED. The dynamic routing algorithm is applied to the CapsNet layer to effectively identify overlapping sound events. Meanwhile, the BiGRU Model is used for sequence modeling to acquire contextual information from audio vector sequences by combining BiGRU with a time-distributed fully connected layer. We evaluated the model on the VWLSD and the DESD. Compared to the baseline system, our ResCapsnet-BiGRU model is shown to exhibit superior performance and robustness.

In the future, we will continue to explore more efficient feature extraction techniques and investigate how to improve dynamic routing capsule networks to better handle dynamic and complex patterns, thereby further improving the accuracy of SED.

Abbreviations

SED	Sound event detection
AT	Audio tagging
CNN	Convolutional neural networks
CRNN	Convolutional recurrent neural networks
CRAM	Customized residual attention module
BiGRU	Bidirectional gated recurrent unit
PSED	Polyphonic sound event detection
HMM	Hidden markov model
NMF	Nonnegative matrix factorization
GMM	Gaussian mixture model
SVM	Support vector machine
DNN	Deep neural networks
LSTM	Long short-term memory
GCCaps	Gated convolution capsule
VWLSD	Vehicle Weakly Labeled Sound Dataset
DESD	Domestic Environment Sound Dataset

Acknowledgements

Not applicable.

Authors' contributions

Bing Sun: conceptualization, formal analysis, investigation methodology, software, writing—original draft. Chenglong Liu: formal analysis, software, writing—original draft. Shuguo Yang: conceptualization, funding acquisition, supervision, writing—original draft. Wenwu Wang: conceptualization, supervision, writing—review editing. Yiduo Mei: data curation, software, validation.

Funding

Not applicable.

Data availability

The datasets supporting the conclusions of this article are available on the internet. The DataSet of DCASE 2017 task 4 was obtained from DCASE Website, dcase.community/challenge2017/download.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 29 January 2025 Accepted: 2 June 2025
Published online: 13 June 2025

References

1. K. Imoto and S. Kyochi, Sound event detection utilizing graph laplacian reg- ularization with event co-occurrence, IEICEe Transactions on Information and Systems **E103.D**, 1971–1977 (2020).
2. J.-L. Rouas, J. Louradour, and S. Ambellouis, Audio events detection in public transport vehicle, in IEEE Intelligent Transportation Systems Conference. IEEE 733–738 (2006)

3. A. Temko, E. Monte, and C. Nadeu, Comparison of sequence discriminant support vector machines for acoustic event classification, in IEEE International Conference on Acoustics Speech and Signal Processing Proceedings. IEEE, **5**, pp. V–V (2006)
4. J. Meng, X. Wang, J. Wang, X. Teng, Y. Xu, A capsule network with pixel-based attention BGRU and for sound event detection. Digital Signal Processing **123**, 103–434 (2022)
5. A. Dang, T. H. Vu, and J.-C. Wang, A survey of deep learning for polyphonic sound event detection, in International Conference on Orange Technologies (ICOT). IEEE, 75–78 (2017)
6. N. Degara, M.E. Davies, A. Pena, M.D. Plumbley, Onset event decod- ing exploiting the rhythmic structure of polyphonic music. IEEE Journal of Selected Topics in Signal Processing **5**, 1228–1239 (2011)
7. J.J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, F.J. Canadas-Quesada, Musical instrument sound multi-excitation model for non-negative spectrogram factorization. IEEE Journal of Selected Topics in Signal Processing **5**, 1144–1158 (2011)
8. T. Heittola, A. Mesaros, A. Eronen, T. Virtanen, Audio context recogni- tion using audio event histograms, in, 18th European Signal Processing Conference. IEEE **2010**, 1272–1276 (2010)
9. G. Guo, S.Z. Li, Content-based audio classification and retrieval by support vector machines. IEEE Transactions on Neural Networks **14**, 209–215 (2003)
10. Y. Liu, J. Tang, Y. Song, and L. Dai, A capsule based approach for polyphonic sound event detection, in 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 1853–1857 (2018)
11. I. McLoughlin, H. Zhang, Z. Xie, Y. Song, W. Xiao, Robust sound event classification using deep neural networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**, 540–552 (2015)
12. M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen, A convolutional neural network approach for acoustic scene classification, in International Joint Conference on Neural Networks (IJCNN). IEEE, 1547–1554 (2017)
13. Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging, in Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017. ISCA, 3083–3087 (2017)
14. H. Zhang, I. McLoughlin, and Y. Song, Robust sound event recognition using convolutional neural networks, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 559–563 (2015)
15. H. Phan, L. Hertel, M. Maass, and A. Mertins, "Robust audio event recognition with 1-max pooling convolutional neural networks," in Proceedings of Interspeech 2016. San Francisco, USA: ISCA, 3653–3657 (2016)
16. E. Cakır, T. Heittola, and T. Virtanen, "Domestic audio tagging with convolutional neural networks," in Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop. Tampere University of Technology. Laboratory of Signal Processing, (2016)
17. H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection, in Proceedings of Interspeech 2022. ISCA, 2763–2767 (2022)
18. J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, Semi-supervsied learning-based sound event detection using freuqency dy- namic convolution with large kernel attention for dcase challenge 2023 task 4, arXiv e-prints, p. arXiv:2306.06461, (2023)
19. S. Xiao, J. Shen, A. Hu, X. Zhang, P. Zhang, Y. Yan, Sound event detection with weak prediction for dcase 2023 challenge task4a. Tech. Rep., DCASE2023 Challenge (2023)
20. S. Sabour, N. Frosst, and G. E. Hinton, Dynamic routing between capsules, in Asia-Pacific Signal and Information Processing Association Annual Sum- mit and Conference (APSIPA ASC). Red Hook, NY, USA: Curran Associates Inc. **30**, 1853–1857 (2018)
21. F. Vesperini, L. Gabrielli, E. Principi, S. Squartini, Polyphonic sound event detection by using capsule neural networks. IEEE Journal of Selected Topics in Signal Processing **13**, 310–322 (2019)
22. T. Iqbal, Y. Xu, Q. Kong, and W. Wang, Capsule routing for sound event detection, in 2018 26th European Signal Processing Conference (EUSIPCO). IEEE, 2255–2259 (2018)
23. A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017). Western Finland, Finland: Tam- pere University of Technology. Laboratory of Signal Processing, 85–92 (2017)
24. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 770–778 (2016)
25. A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Esresnet: Environmental sound classification based on visual domain models," in 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 4933–4940 (2020)
26. Q. Wang, J. Du, H.-X. Wu, J. Pan, F. Ma, C.-H. Lee, A four-stage data augmentation approach to resnet-conformer based acoustic modeling for sound event localization and detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing **31**, 1251–1264 (2023)
27. Z. C. Lipton, J. Berkowitz, and C. Elkan, A critical review of recurrent neural networks for sequence learning, arXiv e-prints, p. arXiv:1506.00019, (2015)
28. A. Graves, A.-r. Mohamed, and G. Hinton, Speech recognition with deep recurrent neural networks, in IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 6645–6649 (2013)
29. G. Parascandolo, H. Huttunen, and T. Virtanen, Recurrent neural networks for polyphonic sound event detection in real life recordings, in IEEE Inter- national Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 6440–6444 (2016)
30. Y. Wang, L. Neves, and F. Metze, Audio-based multimedia event detection using deep recurrent neural networks, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE 2742– 2746 (2016)
31. T. Hayashi, S. Watanabe, T. Toda, T. Hori, J. Le Roux, K. Takeda, Duration-controlled lstm for polyphonic sound event detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**, 2059–2070 (2017)
32. S. Adavanne, A. Politis, J. Nikunen, T. Virtanen, Sound event localiza- tion and detection of overlapping sources using convolutional recurrent neural networks. IEEE Journal of Selected Topics in Signal Processing **13**, 34–48 (2019)
33. Y. Cao, Q. Kong, T. Iqbal, F. An, W. Wang, and M. D. Plumbley, Poly- phonic sound event detection and localization using a two-stage strategy, arXiv e-prints, p. arXiv:1905.00268, (2019)
34. T. Virtanen, A. Mesaros, T. Heittola, A. Diment, E. Vincent, E. Benetos, and B. M. Elizalde, Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017). Tampere University of Technology. Laboratory of Signal Processing (2017)
35. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Computation **9**, 1735–1780 (1997)
36. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, CoRR **abs/1412.3555**, (2014)
37. R. Lu and Z. Duan, Bidirectional GRU for sound event detection, in Detection and Classification of Acoustic Scenes and Events (DCASE) Workshop. Tampere University of Technology. Laboratory of Signal Processing (2017)
38. DCASE 2017 Task4, 2017. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-large-scale-sound-event-detection. Accessed: 12 April 2024
39. N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, Sound event de- tection in domestic environments with weakly labeled data and soundscape synthesis, in Workshop on Detection and Classification of Acoustic Scenes and Events. New York City, United States: Inria (2019)
40. R. Serizel, N. Turpault, A. Shah, and J. Salamon, Sound event detection in synthetic domestic environments, in ICASSP 2020 - 2020 IEEE Inter- national Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE 86–90 (2020)
41. E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, T. Virtanen, Con- volutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**, 1291–1303 (2017)
42. A. Mesaros, T. Heittola, T. Virtanen, Metrics for polyphonic sound event detection. Applied Sciences **6**, 162 (2016)
43. S. Ioffe and C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd In- ternational Conference on Machine Learning. Lille, France: PMLR 37, 448–456 (2015)

44. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv e-prints, p. arXiv:1207.0580 (2012)
45. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research **15**, 1929–1958 (2014)
46. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," (2017). [Online]. Available: https://arxiv.org/abs/1412.6980
47. Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, Large-scale weakly supervised audio classification using gated convolutional neural network, in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE 121–125 (2018)
48. L. Xu, L. Wang, S. Bi, H. Liu, and J. Wang, Semi-supervised sound event detection with pre-trained model, in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE 1–5 (2023)
49. M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**, 2673–2681 (1997)
50. K. Li, S. Yang, L. Zhao, W. Wang, Weakly labeled sound event detection with a capsule-transformer model. Digit. Signal Process. **146**, 104347 (2024)

Publisher's Note