applied sciences

MDPI

Article

Weighted Cluster-Range Loss and Criticality-Enhancement Loss for Speaker Recognition

Jianye Mo and Li Xu *

College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China; jymoee@zju.edu.cn
* Correspondence: xupower@zju.edu.cn;

Received: 8 November 2020; Accepted: 11 December 2020; Published: 16 December 2020

Abstract: While traditional i-vector based methods are popular in the field of speaker recognition, deep learning has recently found more and more applications to the end-to-end models due to its attractive performance. One effective practice is the integration of attention mechanism into the Convolution Neural Networks (CNNs). In this work, a light-weight dual-path attention block is proposed by combining the self-attention and Convolutional Block Attention Module (CBAM), which helps to capture more multi-source features with neglectable extra time expense. Additionally, a Weighted Cluster-Range Loss (WCRL) is proposed to enhance the identification performance of Cluster-Range Loss (CRL) on indecisive samples. Besides, to address the low efficiency in the initial training stage of CRL, a novel Criticality-Enhancement Loss (CEL) is also presented. Both of the proposed loss functions could significantly promote the training efficiency and globally improve the recognition performance. Experimental results are presented to show the effectiveness of the proposed scheme, which achieves a competitive top-1 accuracy of 92.0%, top-5 accuracy of 97.6%, and Equal Error Rate (EER) of 3.5% on the VoxCeleb1 dataset.

Keywords: speaker recognition; attention; loss function

1. Introduction

Generally, there are two subtasks in the field of speaker recognition, i.e., Speaker Identification (SI) and Speaker Verification (SV). SI aims to classify the identity of a given speaker's utterance into the corresponding one among a group of enrolled speakers [1]. Therefore, SI could be treated as a 1:N process. Differing from SI, SV aims to verify whether the identity of the current test speaker is the same as the given enrolled speaker, i.e., it could be treated as a 1:1 process. The concept of speaker recognition can be further divided into two types: text-dependent and text-independent. In this work, the latter one will be explored, which is rather more practical, but is however more difficult to deal with.

For speaker recognition, there have been numerous methods that have been successfully applied to this field. Traditional methods include Vector Quantization (VQ) [2], Gaussian Mixture Models (GMMs) [3,4], and its variant Gaussian Mixture Model-Universal Background Models (GMM-UBMs) [5]. a method with Support Vector Machine (SVM) was also explored [6]. What has dominated the field of speaker recognition are the hybrid models with the i-vector [7–9]. However, it was reported in [10] that the i-vector based approaches are sensitive to the speech duration and may not perform well when dealing with short utterances. Recently, due to the rapid development of deep learning, there has been active research on applying deep learning methods to speaker recognition. Deep Neural Networks (DNNs) are usually utilized to extract frame-level features, followed by an utterance-level feature generating operation like the temporal average. The speaker representations generated by the DNNs are referred to as the d-vector [11]. Based on DNN, the x-vector proposed in [12] employs an isolated PLDA backend to compare the embedding pairs. Actually, employing

deep neural networks to learn the representation for speaker recognition has long been reported [13]. Throughout the recent papers working on speaker recognition, deep neural networks such as Visual Geometry Group Network (VGG-Net) [14] and Residual Network (ResNet) [15] have been frequently utilized. To name only a few, in [8,16,17], VGG-like CNNs were adopted, while ResNet-like networks were also explored in some literature [9,18,19].

Along with the deep neural networks, attention mechanisms have also appealed to many researchers. Various attention mechanisms have been applied in many hot research fields, including Natural Language Processing (NLP) [20–22], speech recognition [8,9,23], and Computer Vision (CV) [24–27]. Self-Attention (SA) is a commonly used attention mechanism and has been successfully integrated into CNNs, especially ResNet [8,9]. Variants of self-attention are also emerging [28–30]. Furthermore, those attention mechanisms considering the spatial and channel dimension are also widely employed especially in the field of CV. Wang et al. [25] proposed a residual attention network where the trunk-and-mask attention module employs a encoder-decoder style. Squeeze and Excitation blocks (SEs) [31] is a channel attention mechanism, where the global spatial information is squeezed by a global average pooling and then the channel-wise dependencies are obtained with fully-connected layers and the sigmoid function. Both in the Convolutional Block Attention Module (CBAM) [26] and Bottleneck Attention Module (BAM) [27], 3D attention map inferences are decomposed into channel and spatial. The channel attention in CBAM achieves better performance than SEs, and to further push the performance, spatial attention is also exploited. In BAM, the 3D attention map is computed along the channel and spatial axis at the same time, while in CBAM, the attention map along these two axes is computed sequentially. Fu et al. [30] proposed a Dual Attention Network (DANet), which contains a position attention module and a channel attention module and also captures global dependencies in the spatial and channel dimensions, respectively.

To achieve satisfactory speaker recognition, a well-designed loss function is also crucial. Traditional softmax loss lacks the ability of discrimination [32], and numerous loss functions have been proposed to address this problem [33–37]. Wang et al. [35] proposed a Large Margin Cosine Loss (LMCL), which minimizes intra-class variance and maximizes inter-class variance by virtue of normalization and cosine decision margin maximization. The famous triplet loss [37] and its variants have achieved satisfactory performance. Yu et al. [38] proposed a Weighted Triplet Loss (WTL), which imposes weights on the terms of the triplet loss. The weights in this loss function are obtained by computation and vary with the time steps. Both weighted triplet loss and weighted triplet loss with only a positive constraint have achieved better performance than weighted triplet loss with only a negative constraint. Besides, negative samples are randomly selected in weighted triplet loss, while such random selection has been reported to be inefficient in recent literature [9,39]. Due to the low efficiency brought by random sample selection, anchor and hard sample selection are taken into consideration in many emerging approaches with triplet loss. In Coupled Cluster Loss (CCL) [39] and end-to-end loss [40], cluster centers take the place of stochastic anchor selection. The Cluster-Range Loss (CRL) [9] emphasizes the hardest positive and the hardest negative of each speaker and focuses on the cluster distribution instead of separate samples.

Drawing on the recent success of self-attention and CBAM, we propose to combine these two attention mechanisms in our work and form a Dual-path Attention (DA) block. Besides, based on the work in [9] and inspired by the weighted-triplet loss [38], a novel loss is proposed here, i.e., Weighted Cluster-Range Loss (WCRL). Additionally, in order to further improve the convergence efficiency of CRL, we also design another loss function, Criticality-Enhancement Loss (CEL), which focuses on the most easily optimized samples per step. With the proposed methods, the training efficiency and classification performance tested on the VoxCeleb1 dataset are significantly improved.

The rest of this paper is organized as follows. In Section 2, the mechanisms of self-attention, the convolutional block attention module, and cluster-range loss are briefly introduced. Section 3 depicts our proposed dual-path attention architecture, weighted cluster-range loss,

and criticality-enhancement loss. The experimental setup and results are given in Section 4. Finally, the conclusions are presented in Section 5.

2. Related Work

In this section, a brief introduction of the Convolutional Block Attention Module (CBAM), Self-Attention (SA), and Cluster-Range loss (CRL) is given.

2.1. Self-Attention

CNNs are adept at capturing local characteristics, but are inefficient when capturing long-range dependencies. However, this is what attention mechanisms are expert at. Given three vectors, Q, K, and V (namely the queries, keys, and values), an attention function maps them to an output. The queries and keys are used to compute a weight matrix, and the output is the weighted sum of values. When adopting dot-product attention, the attention function could be described as the formula below:

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(QK^T\right)V \tag{1}$$

Self-attention is a special case of the attention mechanism, in that Q, K, and V are from the same input. It provides an effective means for capturing long-range dependencies. There is also a multi-head version of self-attention, where the computation of the attention map is firstly divided into several heads, and the final attention map is concatenated from these heads. In this paper, we adopt the single-head version. More details will be given in Section 3.

2.2. CBAM

The Convolutional Block Attention Module (CBAM) [26] is specially designed for convolutional neural networks. Due to its light weight and general nature, the CBAM could be easily integrated into any given CNN architecture. In the CBAM, attention maps are extracted along the channel and spatial axes, while in traditional convolutional operation, the extracted spatial and cross-channel information are blended. Given an input feature map $x \in \mathbb{R}^{C\times H\times W}$, the CBAM sequentially obtains a channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$ of 1D and a spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ of 2D. The computation is finished sequentially through the two modules, i.e., the channel attention module and spatial attention module. Formally,

$$o' = M_c(x) \otimes x \tag{2}$$

$$o'' = M_s\left(o'\right) \otimes o' \tag{3}$$

where $\otimes$ refers to element-wise multiplication and o'' is the final output of the CBAM.

2.2.1. Channel Attention Module

The attention map of channel attention module is computed as follows.

$$\begin{aligned} M_c(x) &= \sigma(MLP(AvgPool(x)) + MLP(Ma\ xPool(x))) \\ &= \sigma\left(W_1\left(W_0\left(x^c_{avg}\right)\right) + W_1\left(W_0\left(x^c_{\max}\right)\right)\right) \end{aligned} \tag{4}$$

where σ denotes the sigmoid function and MLP means the shared one-hidden-layer multi-layer perceptron.

2.2.2. Spatial Attention Module

For the output of spatial attention module, it is calculated as follows.

$$\begin{aligned} M_s(x) &= \sigma\left(f^{7\times7}([AvgPool(x); MaxPool(x)])\right) \\ &= \sigma\left(f^{7\times7}\left(\left[x^s_{\text{avg}}; x^s_{\text{max}}\right]\right)\right) \end{aligned} \tag{5}$$

where σ denotes the sigmoid function and $f^{7\times7}$ refers to a convolutional operation of filter size 7×7. $x^s_{avg} \in \mathbb{R}^{1\times H\times W}$ and $x^s_{ma\ x} \in \mathbb{R}^{1\times H\times W}$ represent features passing through the average pooling layer and the maximum pooling layer, respectively. Both pooling operations are done across the channel axis.

2.3. Cluster-Range Loss

Different from other variants of triplet loss, the cluster-range loss [9] focuses on the cluster distribution of positives and negatives. Shrinking the distribution range of positives, whilst extending the distance on the boundary of positive distribution and its nearest negative distribution are the goals of cluster-range loss.

The requirement of cluster-range loss can be described as below:

$$D^n > D^p + \alpha, \quad \forall(n,p) \in \mathcal{C} \tag{6}$$

$$S^p > S^n + \alpha, \quad \forall(n,p) \in \mathcal{C} \tag{7}$$

where Formula (6) is in the form of the Euclidean distance metric and Formula (7) is in the form of the cosine similarity metric. D^n denotes the minimum inter-class distance, and D^p denotes the maximum intra-class distance. Correspondingly, S^p means the minimum intra-class similarity, and S^n means the maximum inter-class similarity. Following the experience in [41], CRL is formed mainly by the cosine similarity metric.

The exemplar mining method of cluster-range loss is illustrated as Figure 1. For each batch, K speakers are randomly selected with M utterances stochastically picked for each speaker. The hard positive and hard negative for each utterance are selected by virtue of the similarity matrix by the min and max operations. Formally,

$$h^p_{i,j} = \min_{k=1,M} s_{i,j,k} \tag{8}$$

$$h^n_{i,j} = \max_{k=M,..K\times M} s_{i,j,k} \tag{9}$$

where $h^p_{i,j}$ and $h^n_{i,j}$ respectively denote hard positive and hard negative in the given batch for the jth utterance of the ith speaker.

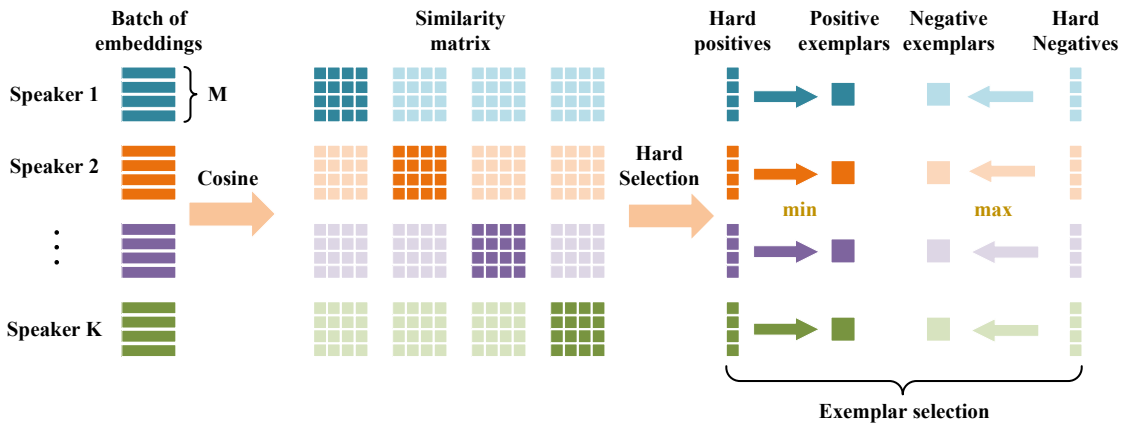


Figure 1. Diagram of the online exemplar mining method for cluster-range loss.

The exemplars are further obtained by the formulas below:

$$e_i^p = \min_{l=1\ldots M} h_{i,l}^p \tag{10}$$

$$e_i^n = \max_{j=1\ldots M} h_{i,j}^n \tag{11}$$

where e_i^p and e_i^n represent positive and negative exemplars in the given batch for the ith speaker, respectively.

Cluster-range loss consists of a hard loss and a normal loss, which could be describe as Formulas (12) and (13). By constructing a mass of positive and negative pairs, the normal loss serves to stabilize the training process. On the contrary, the hard loss centers on the hard positives and negatives and leads to more efficient convergence. m is a hyperparameter that is larger than one, and it makes normal loss dominate the training process in the initial stage.

$$\begin{aligned} L_{\mathrm{CRL}}^{\mathrm{hard}} &= \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[e_i^n - h_{i,j}^p + \alpha \right]_+ \\ &+ \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[h_{i,j}^n - e_i^p + \alpha \right]_+ \end{aligned} \tag{12}$$

$$\begin{aligned} L_{\mathrm{CRL}}^{\mathrm{normal}} &= \frac{1}{K \times (K-1) \times M^2 \times (M-1)} \\ &\times \sum_{i=1}^{K} \sum_{a=1}^{M} \sum_{p=1}^{M-1} \sum_{n=1}^{(K-1)\times M} \left[s_{i,a,n} - s_{i,a,p} + \alpha \right]_+ \end{aligned} \tag{13}$$

$$L_{\mathrm{CRL}} = L_{\mathrm{CRL}}^{\mathrm{hard}} + m \times L_{\mathrm{CRL}}^{\mathrm{normal}} \tag{14}$$

3. Proposed Approaches

In this work, in order to take full advantage of different attention mechanisms, we propose a dual-path attention module, which combines the CBAM and self-attention. Besides, to improve the convergence efficiency of CRL, a weighted cluster-range loss and a criticality-enhancement loss are proposed from different standpoints.

3.1. Dual-Path Attention Module

The self-attention mechanism tries to learn the non-local dependencies, while the channel attention in CBAM is applied globally and aims to learn the inter-channel relationship, and the spatial attention is for local feature capturing and aims at learning the inter-spatial correlation. We propose to combine self-attention and the CBAM, expecting that they can compensate for each other. Different from the operation in [26], in this work, the output feature map of the CBAM is not directly added to the input feature map with a residual connection, but is firstly multiplied by a learnable parameter γ_1, like the operation in [24] for SA GAN. For the self-attention branch, we also take similar measures and employ a learnable parameter γ_2. Both γ_1 and γ_2 are initialized to zero, because in the early stage of training, the previous convolutional layers are not well trained, and the attention map obtained in this stage may not contribute to the training positively. As the training progresses, the attention maps become more meaningful, and both γ_1 and γ_2 increase. The diagram of the proposed dual-path attention is depicted as Figure 2.

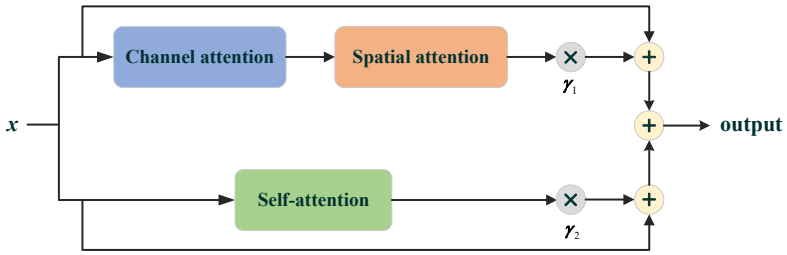


Figure 2. Schematic diagram of dual-path attention.

3.1.1. Convolutional Block Attention Branch

The convolutional block attention module consists of a channel attention module and a spatial attention module. Figure 3 depicts the diagrams of these two modules. The output of CBAM branch z_i^{CBAM} could be described as below, and the computation of o'' is depicted in Section 2.

$$z_i^{CBAM} = \gamma_1 o_i'' + x_i \tag{15}$$

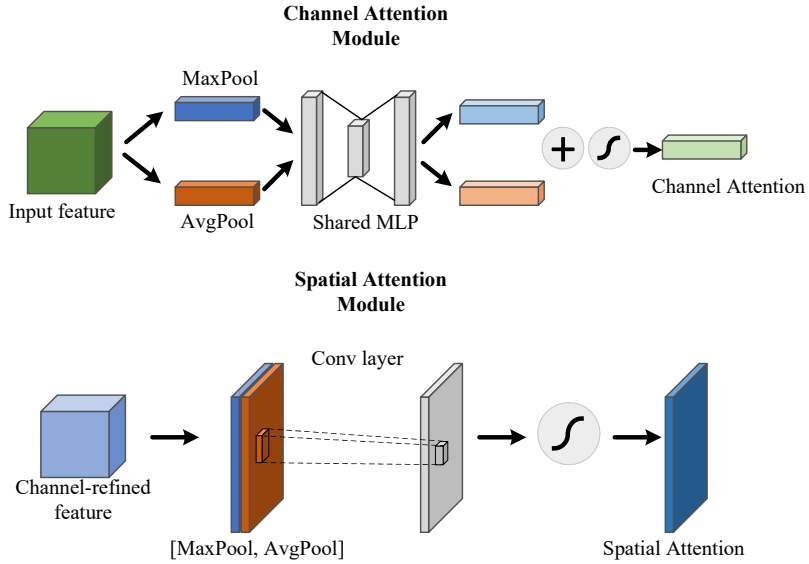


Figure 3. Diagram of the channel attention module and spatial attention module for the Convolutional Block Attention Module (CBAM).

3.1.2. Self-Attention Branch

In this work, we adopt the self-attention introduced in [24]. The computation of this branch is illustrated as Figure 4.

As a special type of attention mechanism, Q, K, and V in the self-attention are all from the same input. Given an input $x \in \mathbb{R}^{C\times H\times W}$, it will be first transformed into three feature spaces, which are named f, g, and h, where $f(x) = W_f(x)$, $g(x) = W_g(x)$, and $h(x) = W_h(x)$. The transformation is done with three 1×1 convolutional units. Then, we have,

$$s_{i,j} = f_i^T g_j \tag{16}$$

$$\beta_{i,j} = \frac{\exp\left(s_{i,j}\right)}{\sum_{j=1}^{N} \exp\left(s_{i,j}\right)} \tag{17}$$

where β is a coefficient matrix of size $N \times N$ ($N = H \times W$), and $\beta_{i,j}$ denotes how important the jth element is for the ith synthesizing response. By the operation below, the output of self-attention is obtained. Formally,

$$o_i''' = \sum_{j=1}^{N} \beta_{i,j} h_j \tag{18}$$

where o''' denotes the output of self-attention. Note that o''' has the same size as x, and thus, they could be added directly. However, as explained above, a trainable parameter γ_2 is utilized here to adjust the ratio of attention in the final output. Finally, the output of the self-attention branch could be describe as below,

$$z_i^{SA} = \gamma_2 o_i''' + x_i \tag{19}$$

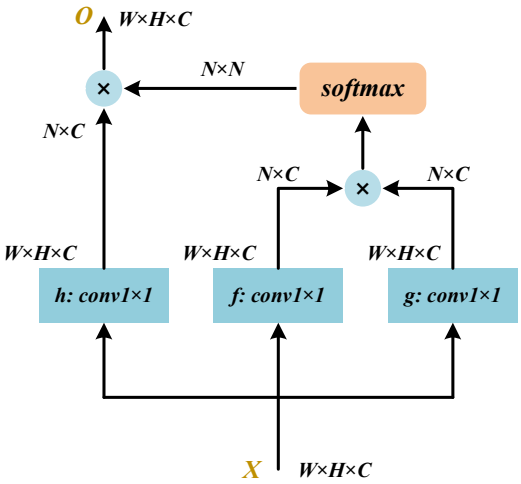


Figure 4. Diagram of the employed self-attention module.

3.1.3. Output of Dual-Path Attention

The attention maps of the CBAM branch and self-attention branch are computed in parallel. That being said, the time consumption of such an operation will be quite similar to that of the architecture solely using either of them. After that, the outputs of these two branches are fused with an add operation. Formally,

$$z_i^{DA} = z_i^{SA} + z_i^{CBAM} \tag{20}$$

where z^{DA} is the final output of dual-path attention module.

3.2. Weighted Cluster-Range Loss

As discussed in [38], the terms in triplet loss may contribute differently to the performance, In this work, we borrow the idea from Weighted Triplet Loss (WTL) [38] into cluster-range loss and propose a comprehensive loss function: Weighted Cluster-Range Loss (WCRL). The only difference between cluster-range loss and weighted cluster-range loss is the weighting factors imposed on the terms of the loss function. Different from WTL, the weighting factors in WCRL are hyperparameters. Like cluster-range loss, the weighted cluster-range loss also consists of two parts, i.e., the hard loss ($L_{\text{WCRL}}^{\text{hard}}$) and the normal loss ($L_{\text{WCRL}}^{\text{normal}}$). For speaker identification, the weighting factor ω_1 is a hyperparameter that is slightly larger than one, and ω_2 is designed to keep one, which we will discuss later. If we set both ω_1 and ω_2 to one, then the weighted cluster-range loss degrades into the standard cluster-range loss.

$$\begin{aligned} L_{\text{WCRL}}^{\text{hard}} = & \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[e_i^n - \omega_2 h_{i,j}^p + \alpha \right]_+ \\ & + \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[\omega_1 h_{i,j}^n - e_i^p + \alpha \right]_+ \end{aligned} \tag{21}$$

$$L_{\mathrm{WCRL}}^{\mathrm{normal}} = \frac{1}{K \times (K-1) \times M^2 \times (M-1)} \times \sum_{i=1}^{K} \sum_{a=1}^{M} \sum_{p=1}^{M-1} \sum_{n=1}^{(K-1)\times M} \left[\omega_1 s_{i,a,n} - \omega_2 s_{i,a,p} + \alpha\right]_+ \tag{22}$$

$$L_{\mathrm{WCRL}} = L_{\mathrm{WCRL}}^{\mathrm{hard}} + m \times L_{\mathrm{WCRL}}^{\mathrm{normal}} \tag{23}$$

Figure 5 illustrates the learning process of weighted cluster-range loss. As mentioned before, the speaker identification task could be treated as a multi-class classification process. As shown in Figure 5, the embeddings of Class 1 ($C1$) and Class 3 ($C3$) are already correctly classified, and thus, it is unnecessary to pay more attention to them. Besides, for embeddings in Class 2 ($C2$) and Class 4 ($C4$), $\boldsymbol{a}$ is the representative of those close to the center of the cluster, while $\boldsymbol{b}$ and $\boldsymbol{c}$ are the representatives of those far away from the center of the corresponding cluster. Intuitively, it is quite easy to correctly classify $\boldsymbol{a}$ into $C1$, while $\boldsymbol{b}$ and $\boldsymbol{c}$ tend to be misclassified into the class of the other. For the classification task, the final output category is the one with the highest score, even if the highest score is actually not that high. Suppose that $S_2 = S_1 + \epsilon$(ϵ is small and $\epsilon > 0$); to correctly classify $\boldsymbol{b}$, we just need to slightly diminish S_2, while it does not matter whether S_1 increases greatly. We argue that $\boldsymbol{b}$ and $\boldsymbol{c}$ will stand a good chance of being correctly classified as long as they get rid of such a critical state. Furthermore, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are essentially different embeddings, and thus, it might be easier to diminish the similarity between $\boldsymbol{b}$ and $\boldsymbol{c}$ than to enlarge the similarity between $\boldsymbol{b}$ and $\boldsymbol{a}$. Thus, we hope to reduce $s_{b,c}$ (namely S_2 in Figure 5) in priority. To this end, a weighting factor ω_1, which is slightly larger than one, is imposed on the loss function. As is shown in Formulas (21) and (22), ω_1 is imposed respectively on $s_{i,a,n}$ and $h_{i,j}^n$. We have also tried the situation where ω_1 is also imposed on e_i^n, which however did not show satisfactory performance. We suspect that this is because e_i^n represents those hardest samples and does not contribute much to the overall classification accuracy. Moreover, we also impose weighting factor ω_2 on the loss function. However, this is only to make our loss function more general in form, and for the moment, we prefer to set ω_2 to one. Both ω_1 and ω_2 are hyperparameters, and the identification performance of different ω_1 and ω_2 will be discussed in Section 4.4.

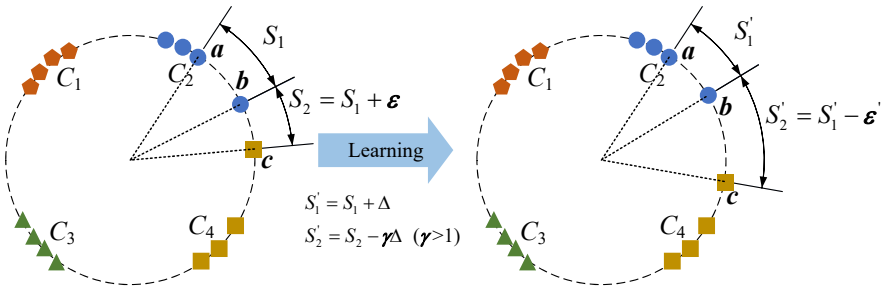


Figure 5. Learning process of the weighted cluster-range loss.

3.3. Criticality-Enhancement Loss

Previous improvements on triplet loss were inclined towards anchor or hard sample selection [9,39,40]. For instance, the hard loss in CRL stresses the selection of the hardest positive and negative for each speaker. Nevertheless, the optimization of the hardest samples is difficult in the initial stage of the training. To address this situation, another novel loss function is proposed, which is referred to as Criticality-Enhancement Loss (CEL). The CEL centers on selecting the most easily, but necessarily optimized triplets (we call them critical samples in this work) and thus dramatically speeds up the training process.

A novel online critical sample mining method is presented for this loss, as illustrated in Figure 6. Again, we need to fetch the batch mentioned in Section 2.3. For each utterance, according to the cosine similarity, its positives are sorted from the largest to the smallest, while its negatives are sorted from the smallest to the largest. To obtain the critical positive and the critical negative, the sorted positives

and sorted negatives are traversed sequentially, and we try to find out the first pair (p-n) where the current $s_{a,n}$ is larger than the current $s_{a,p}$. Then, a loss function is obtained as below,

$$L_{crt} = \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[C_{i,j}^{n} - C_{i,j}^{p} \right]_{+} \tag{24}$$

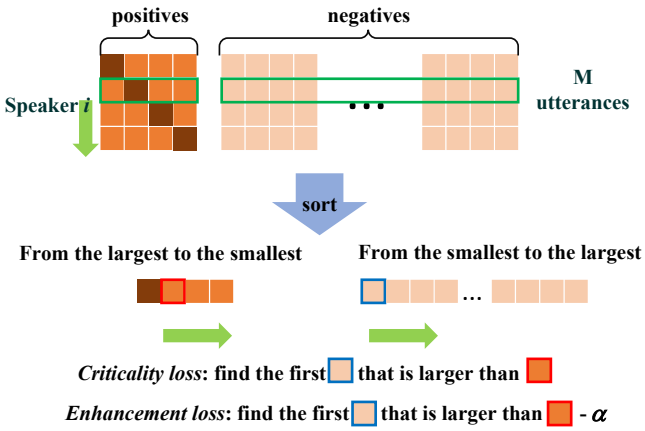


Figure 6. Diagram of the critical sample selection of Criticality-Enhancement Loss (CEL). The values of the dark brown blocks in positives are one and could be skipped. Given the current positive (framed by the red box), the negatives are traversed (represented by the blue box) until the values in the boxes meet the requirement.

Given the jth utterance of the ith speaker as the anchor, L_{crt} tries to find out the most easily optimized hard triplet, where the similarity between anchor and the negative is denoted as $C_{i,j}^{n}$ and the similarity between the anchor and the positive is denoted as $C_{i,j}^{p}$. The max([.], 0) operation is used here in case there is no such critical pair.

The criticality-enhancement loss is proposed to have two parts, and the first part is presented above, i.e., the criticality loss. The criticality loss could push those critical hard samples away from the incorrect state ($s_{a,n}$ is larger than $s_{a,p}$). However, this is insufficient, and we need to push the optimized negatives further away from the optimized positives. Therefore, another part of the proposed loss function is presented, i.e., the enhancement loss. Formally,

$$L_{enh} = \frac{1}{K \times M} \sum_{i=1}^{K} \sum_{j=1}^{M} \left[E_{i,j}^{n} - E_{i,j}^{p} + \alpha \right]_{+} \tag{25}$$

Similar to L_{crt}, L_{enh} aims to find those most easily optimized semi-hard triplets, where the similarity between anchor and the negative are denoted as $E_{i,j}^{n}$ and the similarity between the anchor and the positive is denoted as $E_{i,j}^{p}$. By combining the criticality loss and enhancement loss, the criticality-enhancement loss is formulated as,

$$L_{CEL} = L_{crt} + L_{enh} \tag{26}$$

The CEL aims to degrade the most easily optimized hard triplets per step to semi-hard triplets, whilst degrading the most easily optimized semi-hard triplets to easy triplets. Consequently, the training efficiency is significantly promoted. However, in this work, we still combine it with CRL, in that the optimization of the hard samples dominates the final performance. Formally,

$$L_{total} = L_{CRL} + L_{CEL} \tag{27}$$

The normal loss in CRL serves to stabilize the training, and the hard loss in CRL aims to promote the training effectiveness of the later stage. With the addition of CEL, the training efficiency of the

initial stage is also improved, and relatively more time is gained for the update of the hard loss of CRL. As a result, better performance will be reached.

4. Experiments and Results

4.1. Dataset and Evaluation

In this work, the VoxCeleb1 [16], VoxCeleb2 [19], and CN-Celeb [42] datasets are employed to train and evaluate our models.

In the VoxCeleb1 dataset, there are more than 100,000 utterances coming from 1251 speakers, with around 55% of the speakers being male. These utterances are extracted from videos uploaded to YouTube, covering various accents, ages, and speaking styles. VoxCeleb2 is a much larger dataset containing more than one million utterances from 6112 speakers. Besides, there is no intersection between VoxCeleb1 and VoxCeleb2. When it comes to CN-Celeb, it is a challenging dataset containing 1000 speakers in total and focuses on Chinese celebrities. However, the utterances of some speakers are too few to train, and thus, we only select those speakers with more than 40 speech utterances. Finally, there are 810 speakers in the filtered CN-Celeb dataset. Most of the experiments in this work are conducted on the VoxCeleb1 dataset, while VoxCeleb2 is only used for further evaluating the model for speaker verification, and likewise, the CN-Celeb is merely employed in the speaker identification task.

For the speaker identification task, both training and testing are performed on the same Person Of Interest (POI). The development and test set statistics for VoxCeleb1 and CN-Celeb are presented in Tables 1 and 2, respectively. Top-1 and top-5 accuracy are utilized to evaluate the identification performance. For the speaker verification task, the Equal Error Rate (EER) is employed as the metric to evaluate the verification performance. The development and test set statics for this part are given in Tables 3 and 4. Note that throughout the speaker verification experiments, the test set remains the same, i.e, the test set from VoxCeleb1, and only the training set is altered.

Table 1. Development and test set statistics of VoxCeleb1 for identification. POI, Person Of Interest.

Set	POIs	Utterances
Dev	1251	145,265
Test	1251	8251
Total	1251	153,516

Table 2. Development and test set statistics of CN-Celeb for identification.

Set	POIs	Utterances
Dev	810	113,597
Test	810	12,630
Total	810	126,227

Table 3. Development and test set statistics of VoxCeleb1 for verification.

Set	POIs	Utterances
Dev	1211	148,642
Test	40	4874
Total	1251	153,516

Table 4. Development and test set statistics of VoxCeleb2 for verification.

Set	POIs	Utterances
Dev	5994	1,092,009
Test	118	36,237
Total	6112	1,128,246

4.2. Input Feature

Following the experience in [9], we adopt log Filter-bank (Fbank) coefficients as the input feature for the neural network. Given an utterance from a speaker, it is first framed by the Hamming window with a duration of 25 ms and a shift of 10 ms. After that, silent frames are removed, and we extract log Fbank coefficients for each frame with the dimension being 64. Then, a fragment of continuous 320 frames is randomly selected from the sequence. Thus, for each utterance, an input feature of 320×64 is given.

4.3. Training Methodology

The basic training configuration in this work is similar to that in [9]. The neural architecture of this work is presented in Table 5. To train the neural networks, we employ the SGD optimizer for each model with a momentum of 0.99. Each configuration will be trained for 100 epochs in total.

Table 5. The Residual (Res)-Dual-path Attention (DA) architecture. SI, Speaker Identification.

Layer Name	Kernel Size	Strides	Output Shape
Conv1	$3 \times 3, 64$	1×1	$320 \times 64 \times 64$
Max pool	$3 \times 3, 64$	2×2	$160 \times 32 \times 64$
Res1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix}$	1×1	$160 \times 32 \times 64$
Conv2	$3 \times 3, 128$	1×1	$160 \times 32 \times 128$
Max pool	$3 \times 3, 128$	2×2	$80 \times 16 \times 128$
Res2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix}$	1×1	$80 \times 16 \times 128$
Conv3	$3 \times 3, 256$	2×2	$40 \times 8 \times 256$
Res3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	1×1	$40 \times 8 \times 256$
Conv4	$3 \times 3, 512$	2×2	$20 \times 4 \times 512$
DA1	$/, 512$	1×1	$20 \times 4 \times 512$
Conv5	$3 \times 3, 512$	2×2	$10 \times 2 \times 512$
DA2	$/, 512$	1×1	$10 \times 2 \times 512$
Average	/	/	512
Length Norm	/	/	512
FC (for SI)	512×1251	/	1251

To begin with, experiments for speaker identification are conducted. To explore the effect of the dual-path attention, we first test the architecture with the dual-path attention module and standard cluster-range loss. To explore the effect of our proposed loss functions, we replace the CRL with WCRL and CRL+CEL, respectively. Besides, we set the weighting factor of WCRL to different values so as to find the best weighting configuration for the model. All the mentioned loss functions are employed jointly with the softmax cross-entropy loss. The latter is only used for training a fully-connected (FC) classifier, while the former is utilized to train the speaker embeddings. To this end, the gradient from the fully-connected layer into speaker embedding is cut, and thus, the speaker embeddings are only updated by the proposed loss functions. The experiments are mainly conducted on the VoxCeleb1 dataset, and the CN-Celeb dataset is further employed to prove the effectiveness of our proposed approaches.

As for speaker verification, considering that the WCRL is specially designed for the identification task, we only evaluate the dual-path attention module and CEL in this part. Experiments are first conducted on VoxCeleb1, and the configuration achieving the best performance will be further applied to VoxCeleb2. Note that the FC layer in the neural architecture is removed during this stage.

4.4. Results

The speaker identification results on VoxCeleb1 in this work will be compared with the baselines listed in Table 6. Let the Residual (Res)-SA architecture with CRL in [9] be the major baseline. In the first experiment, the self-attention in [9] is replaced with our proposed dual-path attention, and we achieve a top-1 accuracy of 90.0% and top-5 accuracy of 96.3%. Next, to explore the effect of the weighted cluster-range loss, we keep $\omega_2 = 1$ whilst experimenting with different ω_1. It turns out that when employing the Res-DA architecture, the best performance is achieved when ω_1 is 1.0004 with the top-1 accuracy being 92.0% and the top-5 accuracy being 97.6%. Besides, all the results of weighted cluster-range loss are better than that of cluster-range loss. The results of different ω_1 are listed in Table 7. To evaluate the criticality-enhancement loss, we replace the loss functions mentioned above with CEL+CRL. It turns out that better result is reached using the Res-SA architecture, with the top-1 accuracy being 90.8% and the top-5 accuracy being 97.0%.

Table 6. Different baselines for speaker identification.

Model	Description
i-vector + SVM [16]	This implementation of this method is accompanied by the release of the VoxCeleb1 dataset.
i-vector/PLDA + SVM [16]	Most of this approach is the same as that of i-vector + SVM, and the only difference is that the PLDA score function is adopted.
VGG-like CNN [16]	This approach employs a VGG-like CNN architecture and takes as input spectrograms of dimension 512×300.
VGG-like CNN + SA [8]	The biggest difference between this approach and the former is the employment of self-attention.
ResNet-18-SA [8]	The frameworks in [8] extended two representative CNNs, i.e., VGG and ResNet, by the self-attention mechanism. There is only one self-attention layer in these two frameworks, and it is placed after the convolutional layers. The ResNet-18 + SA in this work achieves state-of-the-art performance.
Res-SA + CRL [9]	This system integrates self-attention into the residual network. Besides, cluster-range loss was employed as the loss function in [9], which significantly improved the speaker identification performance. The biggest differences between this framework and our work are the attention block and the loss function. The superiority of our framework compared with Res-SA + CRL will be reported in Section 4.4.
VGG-like CNN + CL [17]	Sarthak et al. [17] proposed two VGG-like CNN architectures and named them Network-A and Network-B. Besides, their models were trained jointly using softmax loss and center loss (CL). For speaker identification, Network-B achieved the best performance in their work.
ResNet-34 + LDE [18]	In this system, a Learnable Dictionary Encoding (LDE)-based pooling is combined with a ResNet-34 network. Similar to our work, they also employed 64-dimensional Filter-bank (Fbank) coefficients as the input feature.

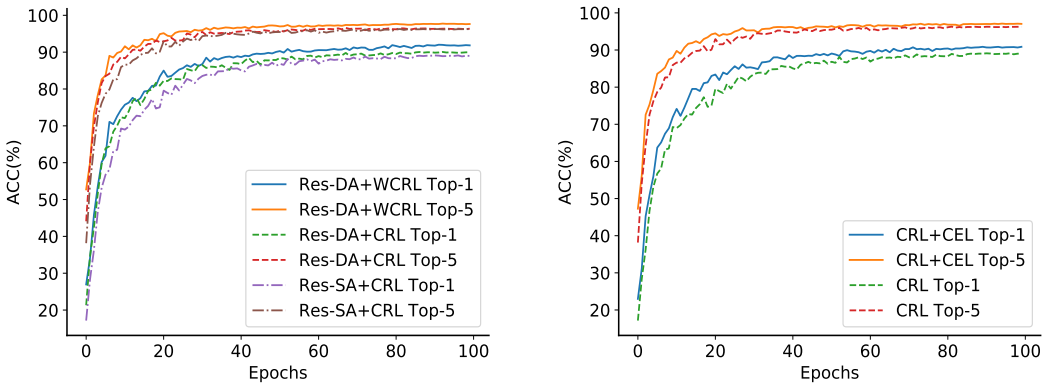
Table 7. Speaker identification results on the VoxCeleb1 dataset with different ω_1 while keeping $\omega_2 = 1$. WCRL, Weighted Cluster-Range Loss.

Accuracy	ω_1	Top-1%	Top-5%
Res-DA + WCRL	1.0006	90.9	96.9
Res-DA + WCRL	1.0005	91.2	97.3
Res-DA + WCRL	1.0004	92.0	97.6
Res-DA + WCRL	1.0003	91.2	97.2
Res-DA + CRL	1.0000	90.0	96.3

When compared with the Res-SA architecture, by employing our dual-path attention module, the top-1 accuracy increases by 0.9% and the top-5 accuracy by 0.5%. Besides, we found that such an operation did not bring extra perceptible computational burden owing to our dual-path architecture, i.e., the attention maps of both branches are calculated concurrently. By applying the idea of weighting, the proposed WCRL achieves significant improvement over the standard CRL and further improves the top-1 accuracy by 2.0% and the top-5 accuracy by 1.3% absolutely. Finally, our best results surpass the baseline (Res-SA + CRL) by 2.9% on the top-1 accuracy and 1.8% on the top-5 accuracy. When it comes to CEL, the result of using SA is slightly better than that of using DA, improving the top-1 accuracy by 1.7% and the top-5 accuracy by 1.2% compared with the baseline. Figure 7 depicts the accuracy curves of our proposed approaches and the baseline Res-SA + CRL. Obviously, the curves of our proposed methods lie totally above that of the baseline, which proves the effectiveness of our methods. The best result given in [8] achieved a top-1 accuracy of 90.8%, while our best model further improves on their best performance by 1.2%. The comparison between our results and other models is shown in Table 8.

Table 8. Speaker identification results on the VoxCeleb1 dataset (higher is better).

Accuracy	Top-1%	Top-5%
i-vector + SVM [16]	49.0	56.6
i-vector/PLDA + SVM [16]	60.8	75.6
VGG-like CNN [16]	80.5	92.1
Res-SA [9]	85.5	93.9
VGG-like CNN + SA [8]	88.2	93.8
VGG-like CNN + CL [17]	89.5	97.0
ResNet-34 + LDE [18]	89.9	95.7
ResNet-18 + SA [8]	90.8	96.5
Res-SA + CRL[9]	89.1	95.8
Res-CBAM + CRL (ours)	89.5	96.0
Res-DA + CRL (ours)	90.0	96.3
Res-DA + CRL + CEL (ours)	90.2	96.7
Res-SA + CRL + CEL (ours)	90.8	97.0
Res-SA + WCRL (ours)	91.8	97.5
Res-DA + WCRL (ours)	92.0	97.6



(**a**) SA vs. DA, CRL vs. WCRL (**b**) CRL vs. CRL+CEL, both using SA

Figure 7. ACC curves of different configurations. CEL, Criticality-Enhancement Loss.

Furthermore, we also tested the cases where the weighting factor ω_2 is not one. The results of this part are presented in Table 9. Note that ω_1 is imposed on h^n and $s_{a,n}$, and ω_2 is imposed on h^p and $s_{a,p}$. Keeping $\omega_1 = 1$, no matter $\omega_2 > 1$ or $\omega_2 < 1$, the results are worse than the version of $\omega_1 > 1$ and $\omega_2 = 1$. When we set $\omega_1 < 1$, the result is also unsatisfactory. Besides, we also evaluate the

situation where $\omega_1 > 1$ and $\omega_2 < 1$, and it turns out that the identification accuracy is similar to other suboptimal situations mentioned above.

Table 9. Speaker identification results on the VoxCeleb1 dataset with different weight positions (higher is better).

ω_1	ω_2	Top-1%	Top-5%
1	1.0005	89.3	96.3
1	0.9995	89.9	96.7
1.0004	0.9995	89.4	96.5
0.9995	1	89.2	96.3
1.0004	1	92.0	97.6
1	1	90.0	96.3

The CN-Celeb dataset was employed to further evaluate our methods, and the results of this part are shown in Table 10. The method in [9] (Res-SA + CRL) was first tested, and a top-1 accuracy of 81.3% and a top-5 accuracy of 90.8% are obtained. The use of dual-attention increases the Top-1 accuracy to 82.6%. Let ω_1 in WCRL be 1.0002 with ω_2 being one; a top-1 accuracy of 83.6% is obtained. Finally, by replacing the WCRL with CRL+CEL, a top-1 accuracy of 84.3% is reached.

Table 10. Speaker identification results on the CN-Celeb dataset.

Accuracy	Top-1%	Top-5%
Res-DA + CRL + CEL	84.3	92.1
Res-DA + WCRL($\omega_1 = 1.0002$)	83.6	91.5
Res-DA + CRL	82.6	91.4
Res-SA + CRL	81.3	90.8

The results of speaker verification are presented in Table 11 and will be compared with the baselines listed in Table 12. When trained on VoxCeleb1, the best result was reached by the scheme of Res-DA + CRL + CEL, and it achieved an EER of 5.1%. To further evaluate the performance of the model on other datasets, the training set was replaced with VoxCeleb2, and the EER was further reduced to 3.5%. Compared with the methods using the i-vector or x-vector, our proposed approaches are very competitive.

Table 11. Speaker verification results on the Voxceleb1 dataset (lower is better).

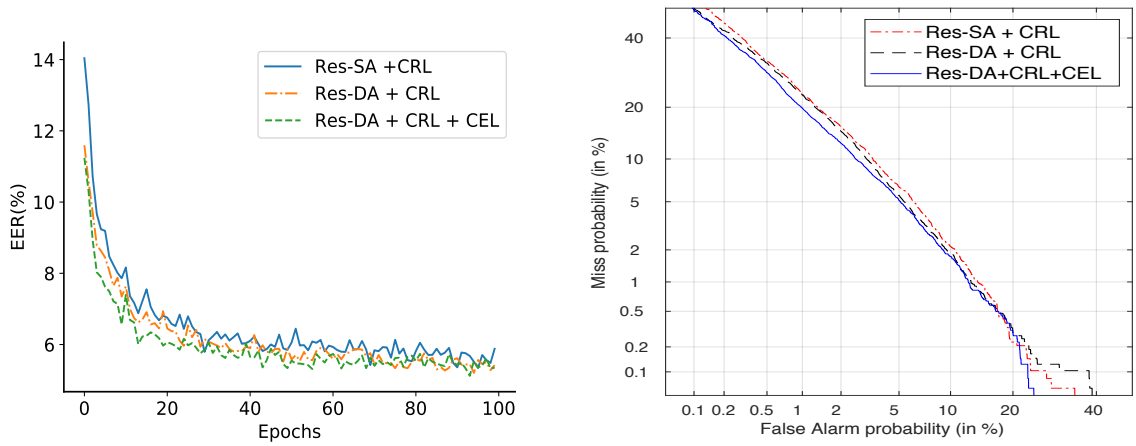
Method	Training Set	EER%
GMM-UBM [16]	VoxCeleb1	15.0
x-vector(cosine) [43]	VoxCeleb1	11.3
i-vector-400/PLDA [16]	VoxCeleb1	8.8
VGG-M [16]	VoxCeleb1	7.8
x-vector(PLDA) [43]	VoxCeleb1	7.1
Res-SA + CRL [9]	VoxCeleb1	5.5
i-vector-2048/PLDA [43]	VoxCeleb1	5.4
Res-DA + CRL (ours)	VoxCeleb1	5.2
Res-DA + CRL + CEL (ours)	VoxCeleb1	5.1
VGG-M [16,19]	VoxCeleb2	5.9
ResNet-34 [19]	VoxCeleb2	5.0
ResNet-50 [19]	VoxCeleb2	4.2
Res-SA + CRL [9]	VoxCeleb2	4.0
Res-DA + CRL + CEL (ours)	VoxCeleb2	3.5

Table 12. Different baselines for speaker verification. UBM, Universal Background Model; CRL, Cluster-Range Loss.

Model	Description
GMM-UBM [16]	In this system, MFCCs of 13 dimensions are employed with the Cepstral Mean and Variance Normalization (CMVN) as the features. A 1024 component UBM is trained for 10 epochs.
i-vectors/PLDA [16,43]	Both the i-vector/PLDA methods using 400 Gaussian components [16] and 2048 Gaussian components [43] for the GMM-UBM are compared.
VGG-M [16]	a CNN architecture proposed by VoxCeleb based on VGG-Medium(VGG-M), with fewer parameters.
x-vector [12]	a famous DNN-based method for speaker verification; a PLDA backend is required to achieve better performance [43].
ResNet-34/50 [19]	Reported with the release of VoxCeleb2; a contrastive loss [44] is employed to train the model.
Res-SA + CRL [9]	See Table 6

4.4.1. Evaluating the Res-DA Architecture

In speaker identification, the proposed Res-DA architecture achieves a best top-1 accuracy of 92.0% on VoxCeleb1. Compared with Res-SA + CRL, when solely replacing the self-attention with CBAM, the top-1 accuracy is 89.5%, which is 0.4% higher than that with self-attention. However, when employing dual-path attention, the top-1 accuracy increases by 0.9%, compared with the self-attention version and increases by 0.5% compared with the CBAM version. When evaluating on CN-Celeb, the use of dual-attention improves the top-1 accuracy by 1.3%, compared with using SA alone. For speaker verification, the use of dual-attention alone reduces the EER by 0.3% on VoxCeleb1. As shown in Figure 8a, the EER curve of DA lies basically below that of SA. All of the above proves the effectiveness of our proposed dual-attention on training.



(**a**) EER curves of different configurations (**b**) DET curves of different configurations

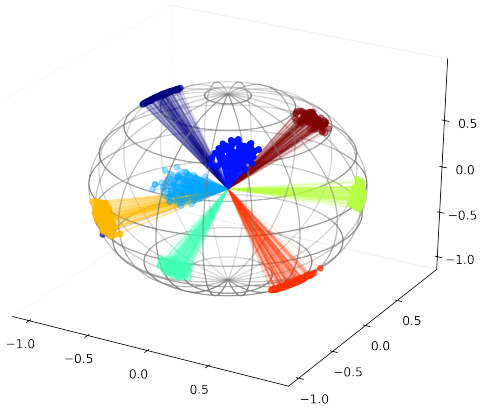
Figure 8. Metric curves for speaker verification. EER, Equal Error Rate; DET, Detection Error Tradeoff.

The CBAM delves into the informative features along the channel and spatial axes. The channel attention module aims to learn "what" is important, and the spatial attention module aims to find out "where" it is informative. Self-attention computes the attention map in a straightforward way and aims at directly learning the dependencies between a pixel itself and other pixels. These three attention mechanisms obtain attention maps from different aspects. During the training, it was found that the addition of the CBAM greatly increased the training efficiency especially in the initial stage. By combining these two attention modules, more informative features are captured, and we achieve

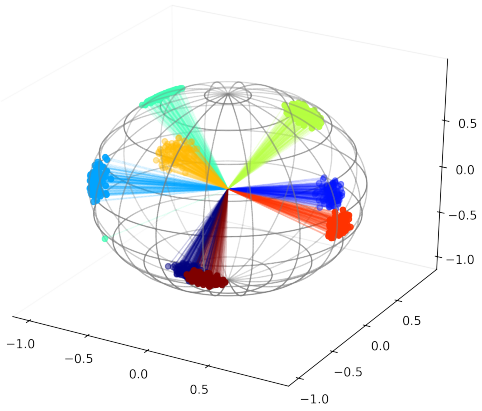
better performance than solely employing either of them in most experiments. Furthermore, due to the light-weight nature of the CBAM and the dual-path architecture, such a combination did not bring obvious additional burden on time, compared with the architecture with self-attention alone.

4.4.2. Evaluating Weighted Cluster-Range Loss

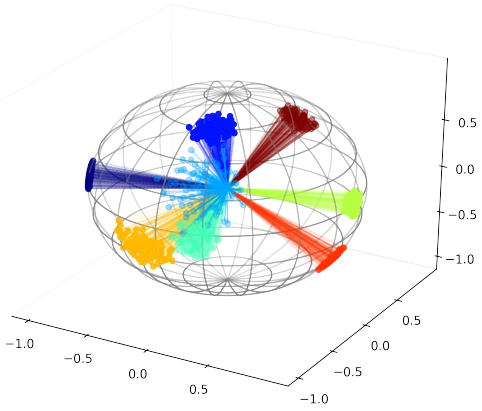
The only difference between cluster-range loss and weighted cluster-range loss is the weighting factors imposed on the loss function. Intuitively, cluster-range loss is a special case of weighted cluster-range loss, with all the weighting factors being one. Weighted cluster-range loss absorbs the advantages of weighted triplet loss and cluster-range loss. The cluster-range loss focuses on the hardest samples, while the idea of weighted triplet loss is to impose different weights on the terms of the loss function. By combining these two ideas, our proposed weighted cluster-range loss could not only converge in a more efficient way by hard sample selection, but also pay more attention to the negative samples and thus help to correctly classify those indecisive embeddings. As is presented in (a) and (c) of Figure 9, some clusters of WCRL have a larger range than that of CRL (especially the cluster in light blue), due to the bias-weight of WCRL. Finally, when using the Res-DA architecture, the weighted cluster-range loss further improves the top-1 accuracy of cluster-range loss by 2.0% on the VoxCeleb1 dataset and by 1.0% on the CN-Celeb dataset.



(**a**) Visualization of features learned with CRL



(**b**) Visualization of features learned with CRL+CEL



(**c**) Visualization of features learned with WCRL ($\omega_1 = 1.0004, \omega_2 = 1$)

Figure 9. Visualization of features learned with different loss configurations.

By imposing the weighting factor, the networks will prioritize the decreasing of cosine similarities between anchors and negatives, and those indecisive samples will be more likely to be pushed

away from the incorrect categories. When applying $\omega_1(>1)$ and $\omega_2(>1)$ simultaneously or solely applying $\omega_2(>1)$, although $s_{a,n}$ has a relative higher weight than that of $s_{a,p}$, the speed of shrinking the distance of positives gets lower, which is however also important for the identification performance. Therefore, it seems that keeping $\omega_2 = 1$ and only changing the value of $\omega_1(>1)$ may lead to better performance. Note that there might be an optimal value for ω_1, but it is still obtained by trial and error for the moment. Let $\omega_1 = 1 + \eta$, then η is a relatively small number and is suggested to be 0.0002–0.0005. There is a subtle balance between the decrease of $s_{a,n}, h^n$ and the increase of $s_{a,p}, h^p$, and thus, ω_1 should not be set too large. The results shows the effectiveness of weighting, while also indicating that imposing a positive-suppressed or positive-attentive weighting factor, i.e., $\omega_2 \neq 1$, may not lead to a satisfactory result.

4.4.3. Evaluating Criticality-Enhancement Loss

Given the current utterance as the anchor, the CEL tries to find the most easily optimized hard and semi-hard triplet. By combing the CRL and CEL, the hardest samples and the most easily optimized hard, semi-hard samples are considered concurrently. Considering a single batch in the CRL + CEL update, each step of the CEL update will push away the negatives mixed in the positive cluster, and the CRL pushes the negatives further away from the positive cluster whilst shrinking the cluster range. Features learned with CRL and CRL+CEL are presented in (a) and (b) of Figure 9, and it turns out that the cluster ranges of the two are similar. The ACC curves of CRL and CRL+CEL (both using the Res-SA architecture) are presented in Figure 7b. Compared with only using CRL, jointly employing CRL and CEL increases the ultimate top-1 accuracy by 1.7% on VoxCeleb1 with SA and by also 1.7% on CN-Celeb with DA. The EER curves and Detection Error Tradeoff (DET) curves are illustrated in Figure 8. Obviously, at most of the points, both the EER curve and the DET curve of CEL+CRL lie lower than that of other configurations.

Compared with WCRL, the CEL alleviates the need for the selection of the weighting factor, and it indeed improves the performance compared with the baseline. Interested readers could add CEL in their loss function provided that they also use triplet-like loss, and we have every reason to believe that this will have a positive effect on their work.

5. Conclusions

In this paper, a light-weight dual-path attention module and two novel loss functions are proposed for text-independent speaker recognition. By combining the CBAM and SA, the model achieves better performance than either self-attention or CBAM only in most experiments, whilst bringing quite small extra burden on the training time. To enhance the performance of CRL, two improvement schemes are respectively presented. First, a comprehensive version of CRL is proposed for the speaker identification task, i.e., the weighted cluster-range loss. By slightly increasing the weight of $s_{a,n}$ and h^n in the loss function, the proposed weighted cluster-range loss could speed up pushing those indecisive samples towards the correct region of classification and serves to converge in a more efficient way. No complex operation, but only a simple hyperparameter is added, and we achieve significant improvement compared with the baseline. In addition, the proposed criticality-enhancement loss centers on the most easily optimized samples. By combining CRL and CEL, both the hardest samples and the most easily, but necessarily optimized samples are considered concurrently. Both theoretical and experimental results prove the superiority of this loss function. Our methods could be generalized to other similar tasks using CNNs and triplet loss. Future work will focus on how to combine the proposed two loss functions and further optimize the model structure.

Author Contributions: Conceptualization, investigation, and writing, original draft preparation and editing, J.M.; writing, review and editing, L.X. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the Robotics Institute of Zhejiang University under Grant No. K11801.

Acknowledgments: The authors would like to thank T. Bian for his contribution to the basic work of the project and the support from Robotics Institute of Zhejiang University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Togneri, R.; Pullella, D. an overview of speaker identification: Accuracy and robustness issues. *IEEE Circuits Syst. Mag.* **2011**, *11*, 23–61. [CrossRef]
2. Soong, F.K.; Rosenberg, A.E.; Juang, B.H.; Rabiner, L.R. Report: a vector quantization approach to speaker recognition. *AT T Tech. J.* **1987**, *66*, 14–26. [CrossRef]
3. Reynolds, D.A. Speaker identification and verification using Gaussian mixture speaker models. *Speech Commun.* **1995**, *17*, 91–108. [CrossRef]
4. Reynolds, D.A.; Rose, R.C. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. Speech Audio Process.* **1995**, *3*, 72–83. [CrossRef]
5. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker verification using adapted Gaussian mixture models. *Digit. Signal Process.* **2000**, *10*, 19–41. [CrossRef]
6. Campbell, W.M.; Campbell, J.P.; Reynolds, D.A.; Singer, E.; Torres-Carrasquillo, P.A. Support vector machines for speaker and language recognition. *Comput. Speech Lang.* **2006**, *20*, 210–229. [CrossRef]
7. Dehak, N.; Kenny, P.J.; Dehak, R.; Dumouchel, P.; Ouellet, P. Front-end factor analysis for speaker verification. *IEEE Trans. Audio Speech Lang. Process.* **2010**, *19*, 788–798. [CrossRef]
8. An, N.N.; Thanh, N.Q.; Liu, Y. Deep CNNs with self-attention for speaker identification. *IEEE Access* **2019**, *7*, 85327–85337. [CrossRef]
9. Bian, T.; Chen, F.; Xu, L. Self-attention based speaker recognition using Cluster-Range Loss. *Neurocomputing* **2019**, *368*, 59–68. [CrossRef]
10. Sarkar, A.K.; Matrouf, D.; Bousquet, P.M.; Bonastre, J.F. Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.
11. Lei, Y.; Scheffer, N.; Ferrer, L.; McLaren, M. a novel scheme for speaker recognition using a phonetically-aware deep neural network. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1695–1699.
12. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
13. Konig, Y.; Heck, L.; Weintraub, M.; Sonmez, K. Nonlinear discriminant feature extraction for robust text-independent speaker recognition. In Proceedings of the RLA2C, ESCA workshop on Speaker Recognition and its Commercial and Forensic Applications, Avignon, Spain, April 1998; pp. 72–75.
14. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
15. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
16. Nagrani, A.; Chung, J.S.; Zisserman, A. Voxceleb: a large-scale speaker identification dataset. *arXiv* **2017**, arXiv:1706.08612.
17. Yadav, S.; Rai, A. Learning Discriminative Features for Speaker Identification and Verification. In Proceedings of the Interspeech, Hyderabad, India, 2–6 September 2018; pp. 2237–2241.
18. Cai, W.; Chen, J.; Li, M. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. *arXiv* **2018**, arXiv:1804.05160.
19. Chung, J.S.; Nagrani, A.; Zisserman, A. Voxceleb2: Deep speaker recognition. *arXiv* **2018**, arXiv:1806.05622.
20. Cheng, J.; Dong, L.; Lapata, M. Long short-term memory-networks for machine reading. *arXiv* **2016**, arXiv:1601.06733.
21. Parikh, A.P.; Täckström, O.; Das, D.; Uszkoreit, J. a decomposable attention model for natural language inference. *arXiv* **2016**, arXiv:1606.01933.

22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
23. Chorowski, J.K.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 577–585.
24. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. *arXiv* **2018**, arXiv:1805.08318.
25. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
26. Woo, S.; Park, J.; Lee, J.Y.; So Kweon, I. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
28. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
29. Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Wang, S.; Zhang, C. Reinforced Self-Attention Network: a Hybrid of Hard and Soft Attention for Sequence Modeling. In Proceedings of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018.
30. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
32. Chung, J.S.; Huh, J.; Mun, S.; Lee, M.; Heo, H.S.; Choe, S.; Ham, C.; Jung, S.; Lee, B.J.; Han, I. In defence of metric learning for speaker recognition. *arXiv* **2020**, arXiv:2003.11982.
33. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. In Proceedings of the ICML, New York, NY, USA, 19–24 June 2016; Volume 2, p. 7.
34. Wang, J.; Zhou, F.; Wen, S.; Liu, X.; Lin, Y. Deep metric learning with angular loss. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2593–2601.
35. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
36. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
37. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: a unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
38. Yu, J.; Zhu, C.; Zhang, J.; Huang, Q.; Tao, D. Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *31*, 661–674.
39. Liu, H.; Tian, Y.; Yang, Y.; Pang, L.; Huang, T. Deep relative distance learning: Tell the difference between similar vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2167–2175.
40. Wan, L.; Wang, Q.; Papir, A.; Moreno, I.L. Generalized end-to-end loss for speaker verification. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4879–4883.
41. Li, C.; Ma, X.; Jiang, B.; Li, X.; Zhang, X.; Liu, X.; Cao, Y.; Kannan, A.; Zhu, Z. Deep speaker: an end-to-end neural speaker embedding system. *arXiv* **2017**, arXiv:1705.02304.

42. Fan, Y.; Kang, J.; Li, L.; Li, K.; Chen, H.; Cheng, S.; Zhang, P.; Zhou, Z.; Cai, Y.; Wang, D. CN-CELEB: A challenging Chinese speaker recognition dataset. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7604–7608.
43. Shon, S.; Tang, H.; Glass, J. Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model. In Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 18–21 December 2018; pp. 1007–1013.
44. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 539–546.