electronics

MDPI

Article

Speaker Recognition Based on the Joint Loss Function

Tengteng Feng [1], Houbin Fan [1], Fengpei Ge [2], Shuxin Cao [1] and Chunyan Liang [1,*]

[1] School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China; fengtt12@126.com (T.F.); fhb_sdut@163.com (H.F.); csx0707@126.com (S.C.)
[2] School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; gefengpei@bupt.edu.cn
* Correspondence: liangchunyan_sdut@163.com

Abstract: The statistical pyramid dense time-delay neural network (SPD-TDNN) model makes it difficult to deal with the imbalance of training data, poses a high risk of overfitting, and has weak generalization ability. To solve these problems, we propose a method based on the joint loss function and improved statistical pyramid dense time-delay neural network (JLF-ISPD-TDNN), which improves on the SPD-TDNN model and uses the joint loss function method to combine the advantages of the cross-entropy loss function and the comparative learning of the loss function. By minimizing the distance between speech embeddings from the same speaker and maximizing the distance between speech embeddings from different speakers, the model could achieve enhanced generalization performance and more robust speaker feature representation. We evaluated the proposed method's performance using the evaluation indexes of the equal error rate (EER) and minimum cost function (minDCF). The experimental results show that the EEE and minDCF on the Aishell-1 dataset reached 1.02% and 0.1221%, respectively. Therefore, using the joint loss function in the improved SPD-TDNN model can significantly enhance the model's speaker recognition performance.

Keywords: SPD-TDNN; joint loss function; cross-entropy loss function; comparative learning; robustness

Citation: Feng, T.; Fan, H.; Ge, F.; Cao, S.; Liang, C. Speaker Recognition Based on the Joint Loss Function. *Electronics* **2023**, *12*, 3447. https://doi.org/10.3390/electronics12163447

Academic Editor: Chiman Kwan

Received: 9 July 2023
Revised: 12 August 2023
Accepted: 13 August 2023
Published: 15 August 2023

1. Introduction

Speaker recognition is a process of identifying speakers from their voices. This is an important research direction in speech signal processing and artificial intelligence.

Before the rise of deep learning, the Gaussian mixture model–universal background model (GMM–UBM) [1] and i-vector [2] systems were popular among traditional speaker recognition methods. They model the speech signal, then extract the speaker's feature vectors, and finally classify feature vectors or match them to a specific speaker. As research on deep learning algorithms has deepened, deep neural network models have shown superior performance in speaker recognition. For example, in 2014, Lei et al. [3] proposed the DNN/i-vector speaker recognition model, which combines the deep neural network (DNN) model with the i-vector model. The basic idea is to use the supervised training DNN instead of the traditional universal background model (UBM) to calculate the frame-level posterior probability, and each output node of the DNN is used as a speaker. The DNN/i-vector method offers improved handling of large amounts of data and complex speaker variations compared with the traditional i-vector method. It also uses a DNN to automatically extract speech features, thereby enhancing robustness and generalization capabilities. However, the DNN/i-vector method requires a large amount of training data and computing resources and has certain limitations when dealing with short speech.

To avoid the limitations and complexity of the i-vector process, Google proposed a d-vector [4] method using a DNN to extract features. The d-vector method directly uses a DNN to map the speech signal to a vector representation of a fixed dimension, eliminating the extraction process of i-vector, and having better speaker adaptability and generalization ability.

In recent years, researchers [5] have woven context information into models based on the time-delay neural network (TDNN) to further improve the d-vector method. For example, x-vector [6] introduces a statistical pooling layer, instead of the average pooling layer, in the TDNN structure, enabling the model to transform frame-level features into segment-level features. However, the x-vector system cannot utilize a wider range of temporal contexts. To solve this problem, researchers introduced a channel attention mechanism, information transmission, and an aggregation mechanism based on the x-vector model into the ECAPA-TDNN [7]. Using squeeze and excitation [8], the method adaptively learns the importance of each channel, enhances the model's attention to important channels, and weakens its attention to unimportant channels. The residual connection Res2Net [9] and information transmission mechanism [10,11] are used to learn and integrate the features of different time steps across channels so that the model can better deal with long sequences and improve the model's performance. At the same time, the ECAPA-TDNN model also adopts a deeper neural network structure and optimizes it to further improve the model's performance.

However, the ECAPA-TDNN model still has some defects. For example, the number of channels in the channel attention mechanism needs to be determined in advance, but it is difficult to determine a number suitable for all scenarios. In addition, the information transmission mechanism adopts a residual connection, which may lead to gradient disappearance or gradient explosion, thus limiting the model's depth and performance. To solve these problems, researchers proposed the D-TDNN [12] model, which introduces a dense connection mechanism to connect the features between different layers, which avoids the problem of gradient disappearance and gradient explosion and enables feature modeling at a deeper level. It can better handle the information interaction between long sequences and different channels, and it has better performance and generalization ability. In addition, the D-TDNN model uses segmented convolution to adaptively learn the weight of each convolution kernel, further improving the model's performance.

The D-TDNN model has strong feature extraction ability but weak modeling ability when a difference arises between different speech signals of the same speaker, and it is easily disturbed by environmental changes and other factors. To further solve this problem, researchers proposed the statistical pyramid dense time-delay neural network (SPD-TDNN) [13] model at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2022, which can effectively improve the D-TDNN's modeling ability and robustness. The core idea of the SPD-TDNN model is to introduce a statistical pyramid pooling layer (SPP) [14] and dense connection mechanism to better deal with long sequences and feature information between different layers. Among them, SPP adopts a multi-scale pyramid pooling layer to fuse features of different scales to improve the model's modeling ability for different time scales. The dense connection mechanism can effectively avoid the problem of gradient disappearance and gradient explosion so that the model can model features at a deeper level. Although SPD-TDNN performs well in speaker recognition tasks, it still has some shortcomings, such as difficulty in adapting to large-scale data, difficulty in dealing with imbalances in training data, a high risk of overfitting training data, and insufficient generalization ability.

The purpose of speaker recognition is to distinguish different speakers through speech signals, which is a typical classification problem. Common classification models usually use the cross-entropy loss function for training, but in the speaker recognition task it is necessary to pay attention to both the differences between different speakers and the similarities between the same speakers. Therefore, the use of cross-entropy loss function alone may not achieve this goal. This paper applies the joint loss function to the speaker recognition task. The main work and contributions of this paper can be summarized as follows.

- We propose a speaker recognition model based on joint loss function training. During the model training process, both cross-entropy loss and contrastive learning loss work together, considering the differences between different speakers and the similari-

ties between the same speakers. This allows the model to better learn speaker-specific feature information.

- To better leverage contrastive learning, we improved the SPD-TDNN module, adjusting the position of the BN layer such that the output of the activation function is normalized through the BN layer to better preserve the input's dynamic range, thereby enhancing the model's nonlinear expression capability and generalization ability.

2. Related Works

2.1. Speaker Recognition System Framework

The speaker recognition system mainly includes two stages: training and recognition. The basic framework is shown in Figure 1. In the speaker recognition system, the speech data are preprocessed first, including data augmentation, pre-emphasis, framing, and windowing. For feature extraction, the most commonly used feature extraction methods are the mel-frequency cepstral coefficient (MFCC) [15] and the filter bank (FBank) feature method [16]. In the training stage, the model is trained as a closed-set speaker recognition model with a classification head. The dimension of the classification head is equal to the number of speakers in the training dataset. The cross-entropy loss function is employed, where the loss is computed by applying softmax nonlinearity to the classification head outputs and comparing them with one-hot encoded labels. Once the training has completed, the classification head is removed, and the model is used as a speaker embedding extractor. In the recognition stage, the test speech feature parameters are input into the established model for matching calculation, and the similarity score is obtained. According to the similarity score, the recognition result is obtained.

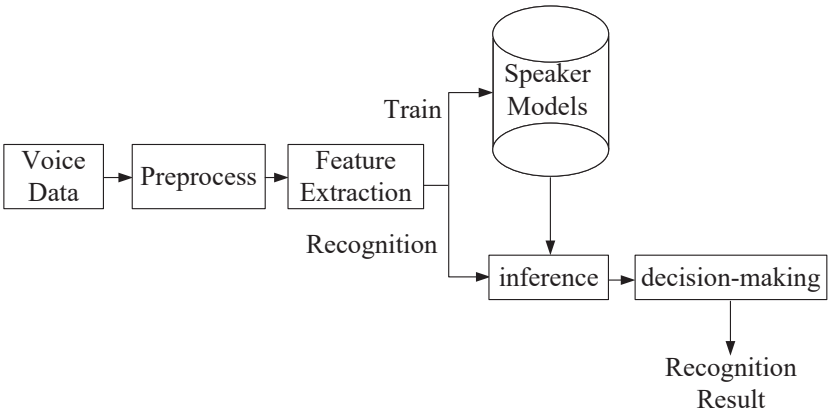

Figure 1. Frame diagram of the speaker recognition system.

2.2. Baseline Network Model

The baseline model used in this paper is SPD-TDNN, and the overall framework is shown in Figure 2. In the multilayer SPD-TDNN module, the output of each layer of the SPD-TDNN module is connected to the subsequent SPD-TDNN layer by using the dense connection mechanism, which realizes information exchange and sharing between layers. At the same time, multiple SPD-TDNN layers are followed by a C-B-R module with a kernel size of 1 to form a SPD-TDNN block. The C-B-R module can be used to aggregate multilayer features from different layers.

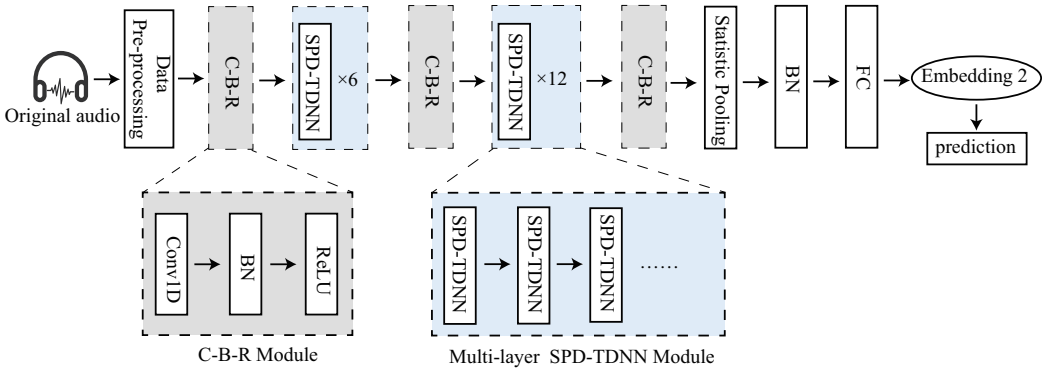

Figure 2. SPD-TDNN overall framework diagram.

2.2.1. SPD-TDNN Layer

The SPD-TDNN layer is the basic unit of the SPD-TDNN model. Each SPD-TDNN layer is composed of the bottleneck layer of the feedforward neural network (FNN) and the SPP. Finally, the input of the SPD-TDNN layer and the output of the SPP layer are connected in series. The SPD-TDNN layer structure is shown in Figure 3, where the dotted box is the SPP layer.

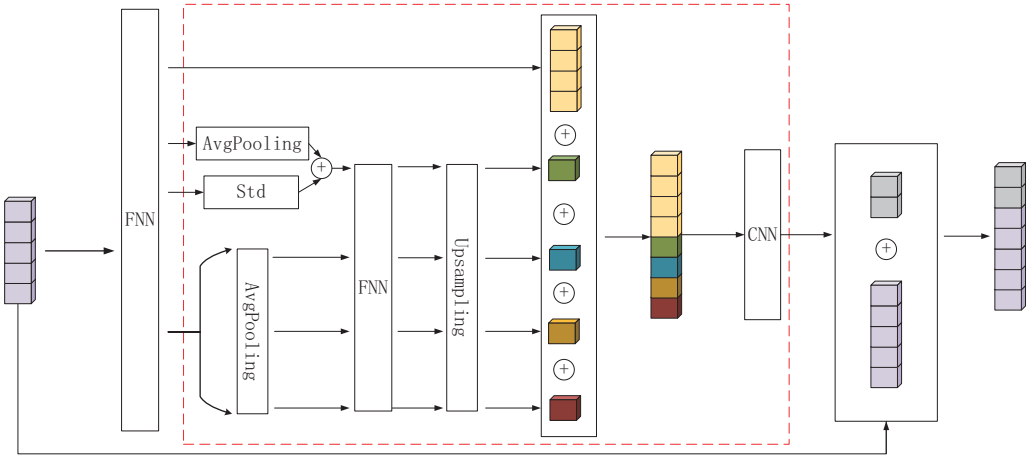

Figure 3. SPD -TDNN layer structure.

The SPP obtains the global and subregional context information from the speech features to capture a more comprehensive feature representation, which helps distinguish the categories of different speakers. First, the input speech features are sent to multiple parallel branches, and the global and subregional context information is collected by establishing the relationship between frames. Then, the outputs of parallel branches are spliced together as fine-grained features of speech. Then, this feature is input into the convolutional neural network (CNN) layer, with the kernel size set to 3, to extract key features. Finally, we combine the obtained key features with the original features to obtain the final features for speaker recognition.

Specifically, the parallel branch includes a global regional branch and multiple subregional branches. The average pooling layer of the global regional branch is a feature that compresses the speech feature into a fixed length in the time domain. To supplement more global context information, we incorporate standard deviation into the global regional branch as a measure of statistical dispersion to reflect the degree of dispersion between individuals in the feature. Finally, the pooled features and the standard deviation are stitched together to form a global feature. The pooling layer of the subregional branch divides the feature map into different regions and forms pooling representations at different locations. To maintain the weight of global features, we add the FNN layer after the pooling

layer, and the dimension represented by the context is changed from N branches to one original dimension.

Finally, the low-dimensional features are up-sampled to the same frame length as the original features by bilinear interpolation and used as the output of parallel branches.

It can be seen from Figure 3 that the number of parallel branches in the SPP layer can be modified. The structure abstracts different subregions by using different sizes of pooling kernels in a few steps. Therefore, the kernel size gap should be maintained to ensure the diversity and complementarity of features. In this paper, the SPP layer contains four parallel branches. Except for the global regional branch, the average pooling layer kernel size used in the subregional branch is set to 1, 16, and 32.

2.2.2. SPD-TDNN Model

The complete network structure of the SPD-TDNN model is shown in Table 1 and consists of four main components.

Table 1. The structure of the baseline network model SPD-TDNN.

	Layer	Structure	Output
1	Conv1D + BN + ReLU	k = 5, p = 2	128
2	SPD-TDNN	(128, 64, 1)	192
	SPD-TDNN	(192, 64, 2)	256
	SPD-TDNN	(256, 64, 3)	320
	SPD-TDNN	(320, 64, 1)	394
	SPD-TDNN	(394, 64, 2)	448
	SPD-TDNN	(448, 64, 3)	512
	Conv1D + BN + ReLU	k = 1, p = 0	256
	SPD-TDNN	(256, 64, 1)	320
	SPD-TDNN	(320, 64, 2)	394
	SPD-TDNN	(394, 64, 3)	448
	SPD-TDNN	(448, 64, 1)	512
	SPD-TDNN	(512, 64, 2)	576
	SPD-TDNN	(576, 64, 3)	640
	SPD-TDNN	(640, 64, 1)	704
	SPD-TDNN	(704, 64, 2)	768
	SPD-TDNN	(768, 64, 3)	832
	SPD-TDNN	(832, 64, 1)	896
	SPD-TDNN	(896, 64, 2)	960
	SPD-TDNN	(960, 64, 3)	1024
	Conv1D + BN + ReLU	k = 1, p = 0	512
3	Statistic Poolong + BN		1024
4	FC + BN		192

(1) In the first component, the CNN layer is used to initialize the number of channels to a fixed size dimension. The convolution kernel size, stride, padding, and dilation are set to 5, 1, 2, and 1, respectively.

(2) In the second component, every three SPD-TDNN layers form a group, and the size of the dilation is set to 1, 2, and 3. The different sizes of expansion and computational efficiency of the parameters are retained so that the model has different receptive fields. These SPD-TDNN groups are combined with the CNN layer to form a total of two SPD-TDNN blocks. The first SPD-TDNN block consists of two SPD-TDNN groups and one CNN layer. The second SPD-TDNN block contains four SPD-TDNN groups and a CNN layer.

(3) The third component is the SPP layer, which aggregates frame-level features and outputs discourse-level features.

(4) The last layer is the fully connected (FC) layer, which is used to output fixed-length speaker embedding vectors.

2.2.3. Other Models

In addition to SPD-TDNN, we also selected D-TDNN, MFA-Conformer, and SE-ResNet as baselines for our comparative experiments.

D-TDNN [12] is a densely connected TDNN layer that effectively reduces the parameter count, enabling fast computation. It serves as a fundamental baseline. MFA-Conformer [17] utilizes conformer blocks as its backbone, combining transformer and convolutional neural network (CNN) elements to capture global and local features efficiently. It stands as a powerful baseline. SE-ResNet [8] focuses on channel relationships, employing squeeze-and-excitation (SE) blocks to explicitly model interdependencies between channels. This adaptive recalibration of channel feature responses showcased state-of-the-art results at that time.

2.3. Data Augmentation

In speaker recognition tasks, data augmentation is usually used to increase the diversity of training data to improve the model's robustness and generalization ability.

2.3.1. Reverberation Enhancement

Reverberation enhancement refers to adding reverberation effects to speech signals to simulate speech signals collected in different recording environments, improving the model's robustness and generalization ability.

In the speaker recognition task, commonly used reverberation enhancement methods use the RIR dataset [18], public reverberation signal libraries (e.g., AURORA-2, AURORA-4 [19]), and reverberation simulators.

2.3.2. Noise Enhancement

Noise enhancement is a common data augmentation method that aims to make the model more robust and able to identify speakers in noisy environments. In speaker recognition, noise datasets are usually used. For example, the MUSAN dataset [20] contains various types of noise signals, which can be added to the original speech signal to increase the noise level and complexity.

2.3.3. SpecAugment

The SpecAugment [21] algorithm is a commonly used spectrum enhancement technology that is primarily used in speech recognition and speech processing tasks, with the basic principle of applying occlusion and distortion transformation on the mel spectrum to generate new training data and thereby improving the model's robustness and generalization ability. Specifically, the SpecAugment algorithm includes three steps.

- Time masking: Randomly select a continuous interval on the timeline and set it to 0, which is equivalent to blocking all the sound signals in the interval. This operation can simulate the interruptions and missing information in the speech signal.
- Frequency masking: Randomly select a continuous interval on the frequency axis and set it to 0, which is equivalent to masking the frequency information of the interval. This operation can simulate noise and distortion in the speech signal.
- Frequency warping: The spectrogram is distorted on the frequency axis to stretch or compress some frequency intervals. This operation can simulate intonation changes and accent differences in speech signals.

During the training process, some SpecAugment operations are randomly applied to generate new training data for each batch. Specifically, each input speech signal is first converted into a mel spectrogram, and then some time masking, frequency masking, and frequency warping operations are randomly applied to generate a new mel spectrogram. Finally, the newly generated mel spectrogram is used as the model input to train the model.

Through these random occlusion and transformation operations, SpecAugment can generate a large amount of diverse training data, which help improve the model's robustness and generalization ability. At the same time, SpecAugment has a low computational cost and does not significantly increase training time and resource consumption.

3. JLF-ISPD-TDNN Model

The framework of the joint loss function and improved statistical pyramid dense time-delay neural network (JLF-ISPD-TDNN) is shown in Figure 4. First, the original audio is preprocessed, including pre-emphasis, extraction of mel features, and the data augmentation methods introduced in Section 2.3 to obtain two sets of enhanced speaker embeddings. Then, the voiceprint features after data preprocessing are input to the improved statistical pyramid dense time-delay neural network (ISPD-TDNN), and effective feature representations are learned through training to obtain Embedding1 and Embedding2. Finally, by using the contrast learning method to maximize the same speaker embedding and minimize the mutual information between different speaker embeddings, we can train a model with better feature representation ability. The C-R-B module in Figure 4 is obtained by placing the BN layer of the C-B-R module in Figure 2 after the convolution layer and the activation function. Among them, ISPD-TDNN is obtained by adjusting the BN layer position of the C-B-R module in the SPD-TDNN model and placing the BN layer after the convolution layer and the activation function. This helps the model better learn useful information in the data, enhance its nonlinear ability, and improve its stability and generalization ability.

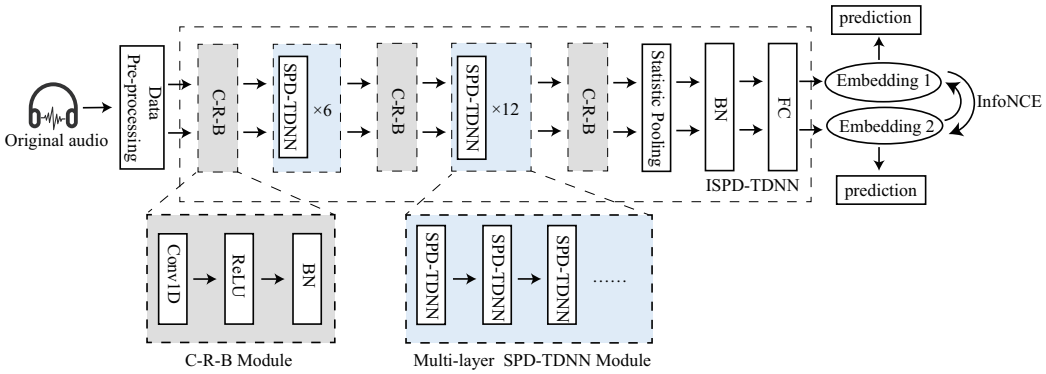

Figure 4. JLF-ISPD-TDNN overall framework diagram.

3.1. AAM-Softmax

The angular additive margin softmax (AAM-softmax) [22] objective function, also known as ArcFace, is a classification loss function based on the angular cosine distance. It is often used in tasks such as face recognition and speaker recognition. Compared with the traditional softmax classifier [23], AAM-softmax enhances the discrimination between samples while ensuring the classification accuracy, which can better solve the problems of unbalanced training data and overlapping between classes.

The core idea of AAM-softmax is to introduce an angle cosine term into the traditional softmax loss function and then use the transformation function of the angle cosine value to adjust the classifier output. This angle cosine term can be regarded as a "margin", which increases the separation of samples in the feature space and increases the distance between different types of samples.

The loss function of AAM-softmax can be expressed as:

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{S(\cos\theta_{yi}+m)}}{e^{s(\cos\theta_{yi}+m)}+\sum_{j=1,j\neq y_i}^{C}e^{s\cos\theta_j}} \quad (1)$$

Among them, N represents the number of samples in the training set, C represents the number of categories, y_i represents the category to which sample i belongs, θ_j represents the angle between the center vector of sample i and category j, s represents the scaling parameter, and m represents the marginal parameter. In each training iteration, the loss function averages the loss of all samples and updates the network parameters by back-propagation.

In general, AAM-softmax optimizes the traditional softmax loss function by increasing the constraint of angular cosine distance, forcing the model to pay more attention to the relative position relationship between samples and improving the classification robustness and accuracy.

3.2. InfoNCE

InfoNCE [24], an algorithm for comparative learning that was originally proposed by Oord et al. [25], is primarily used for feature learning and representation learning in the audio and image fields. Unlike the traditional comparative learning method, the InfoNCE algorithm combines the mutual information in information theory with the neural network, so it can perform effective comparative learning without additional negative samples.

The basic principle of the InfoNCE algorithm is to train a neural network by maximizing mutual information [26] to learn a good feature representation. Mutual information is a measure of the interdependence between two random variables. The larger the value, the stronger the dependence between two random variables. Contrastive learning aims to make the speech signal representations of the same speaker as close as possible and the speech signal representation of different speakers as far away as possible. Therefore, the mutual information between two speech signals can be used as the goal of model training.

Specifically, for a given speech signal, the speech signal is preprocessed and then input to the neural network. The neural network represents the input speech signal as another vector and then determines whether they are from the same speaker by calculating the similarity between the two vectors. The InfoNCE algorithm uses a binary classification problem to train the neural network, where positive samples are speech signals from the same speaker, and negative samples are speech signals from different speakers. The training goal of the model is to maximize the mutual information of positive samples and minimize the mutual information of negative samples [26].

Generally, a batch of speaker embedding pairs $\left\{\left(z_i, z_i^{'}\right)\right\}$ is given, where $\left(z_i, z_i^{'}\right)$ denotes two enhanced speaker embeddings from the same speaker and the batch size is N, which are then fed into the contrast loss function. For positive sample pairs, their similarity is calculated by dot product and softmax normalization, which is expressed as $z_i^T \cdot z_i^{'}$; for negative sample pairs, their similarity is also calculated as $z_i \cdot z_j^{'}$, where i represents the index of positive samples, and j represents the index of negative samples. InfoNCE-based comparative learning loss can be defined as follows:

$$L_{cl} = -\sum_{i=1}^{N} \log \frac{\exp\left(\frac{\mathbf{z}_i^{\top} \mathbf{z}_i'}{\tau}\right)}{\sum_{j=0}^{K} \exp\left(\mathbf{z}_i \cdot \mathbf{z}_j'\right) / \tau} \tag{2}$$

where τ is the temperature hyper-parameter controlling the product sensitivity, and K is the number of negative samples. By minimizing the loss function, we can ensure that the neural network can learn better speech representation, thereby improving the performance of speaker recognition.

3.3. Joint Loss Function

The proposed joint loss function is based on the traditional cross-entropy loss function and incorporates the idea of contrastive learning loss. It considers both the similarity and distinguishability among speech samples to improve the separability and discriminability of

speaker embeddings during training. This way, the model can better differentiate between speech features of different speakers and better distinguish between speech samples of the same speaker.

The joint loss function of the JLF-ISPD-TDNN model consists of two parts: one is the AAM-Softmax function, which is used to classify each sample of the input model; the second is the InfoNCE function, which makes the speaker embedding generated by the model data augmentation more accurate. The joint loss function of model training is as follows:

$$Loss = (1 - \lambda) * L_{AAM} + \lambda * L_{cl} \quad (3)$$

where λ is a hyper-parameter used to balance the weights between the AAM-Softmax loss and the contrastive learning loss.

The goal of the joint loss function proposed in this paper is to minimize the intra-class variance and maximize the interclass variance to improve the separability and discrimination of speaker feature representation. In the model, the model parameters are adjusted by optimizing the joint loss function to obtain a better speaker feature representation.

4. Experiment

4.1. Experimental Setup

The dataset used in the experiment is Aishell-1 [27]. The training set includes 120,421 voices of 340 speakers, and the test set includes 7176 voices of 20 speakers. The Aishell-1 dataset comprises recordings captured in various environments, including both indoor and outdoor settings, along with different background noise conditions. Speech files in this dataset are segmented into short speech segments. Consequently, during training, each sample consists of speech segments from a single speaker, while during testing, each sample contains speech segments from either the same or different speakers for evaluation. In deep learning, data augmentation is beneficial to the training of neural networks because it can increase the diversity of data, enable the model to better adapt to changes in the real world, and improve the model's robustness. Before feature extraction, noise enhancement and reverberation enhancement were performed on each speech. The data augmentation here followed Kaldi's configuration [28] and combined the publicly available MUSAN dataset (music, speech, noise) and RIR dataset (reverberation). Specifically, five enhancement strategies were used: adding reverberation, adding speech, adding music, adding noise, and adding the mixture of speech and music. Each strategy had an equal application probability (0.2) and was randomly selected and applied during the training phase. For the addition of reverberation, we employed the method of discrete linear convolution to evenly mix it into the speech segments of the speaker. This approach ensures that the entire segment incorporates the reverberation effect. Unlike other data augmentation methods, adding reverberation can alter only the "background noise" without changing the "speech content" of the speaker. As for the other data augmentation strategies, we enhanced the original data by adding noise through summation. The proportion of noise was determined using a random function to generate a variety of training data.

The characteristic parameter was an 80-dimensional mel frequency cepstrum coefficient with a frame length of 25 ms and a frame shift of 10 ms, and the feature vector was randomly normalized twice by the cepstrum mean subtraction. After feature extraction, the SpecAugment algorithm was applied to the log mel spectrum as the last enhancement strategy, and the random mask was applied to the time domain and frequency domain of the log mel spectrum. Specifically, we randomly selected a certain moment in the audio, we set the moment and the M frame after the moment to zero, and the frequency domain was the same. By randomly performing mask operations in the time domain and frequency domain, we noted that some speech features were blurred or obscured, resulting in missing local information that would increase the diversity and generalization ability of the data without increasing the data size.

In order to conduct a fair experiment, we used the pytorch framework to implement the proposed ISPD-TDNN and JLF-ISPD-TDNN models. The Adam method was used

to optimize the model, the momentum was set to 0.95, and the weight attenuation was set to 5×10^{-4}. The small batch was set to 128, and the learning rate was initialized to 0.01. For ISPD-TDNN, we use the AAM-Softmax loss functions to classify speakers. For JLF-ISPD-TDNN, we use the AAM-Softmax and InfoNCE loss functions to classify speakers. For AAM-Softmax loss, the margin and scaling parameters were set to 0.2 and 30. For InfoNCE-based comparative learning loss, τ was set to 0.1.

For MFA-Conformer, the optimizer settings are the same as those we proposed. The learning rate is initialized to 0.001. For the multi-head self-attention module, we set the encoder dimension to 256 and the number of attention heads to 4. For the convolutional module, we set the kernel size to 15. For the feed-forward module, we set the number of linear hidden units to 2048. We employed 6 conformer blocks with varying downsampling rates. For SE-ResNet, we also employed Adam as the optimizer with a learning rate initialized to 0.001. We set the base number of channels for SE-ResNet to 32 and the downsampling rate to 1/2. For D-TSDNN, we utilized stochastic gradient descent (SGD) as the optimizer with a momentum of 0.95 and weight decay of 5×10^{-4}. The mini-batch size was set to 128, and the initial learning rate was set to 0.01. For the aforementioned three models, we employed AAM-Softmax loss functions for loss computation.

The evaluation indexes of the experiment were the equal error rate (EER) [29] and minimum detection cost function (minDCF) [28].

4.2. Experimental Results and Discussion

4.2.1. The Impact of λ on Model Performance

Figure 5 shows the impact of the parameter λ on the model's performance as defined in Equation (3). When λ was large, the contribution of InfoNCE became more important, and the model focused more on clustering samples of the same category, which may have led to overfitting. When λ was small, the contribution of the AAM-Softmax classification loss became more important, and the model focused more on the classification ability of different samples of different categories, which may have resulted in the model's inadequate learning of differences between samples of the same category. Therefore, selecting an appropriate value for λ is crucial for the model's performance. The results in Figure 5 show that the InfoNCE loss starts to gradually take effect when λ = 0.2. When λ = 0.4, the model performs optimally, achieving an equal error rate (EER) of 1.02%. However, as λ surpasses 0.5, the model's performance diminishes due to the decreased contribution of the AAM-Softmax loss. In the following experiments, we selected λ = 0.4.

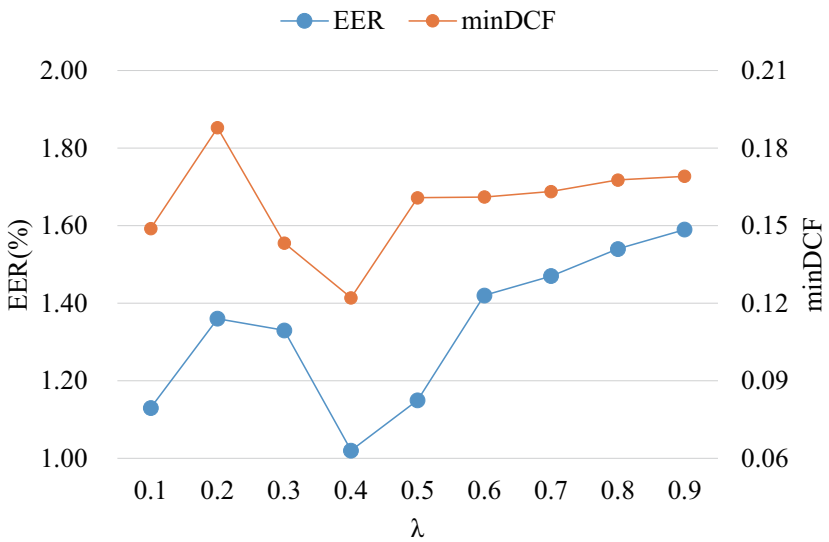

Figure 5. The impact of λ on model performance.

4.2.2. Results and Analysis

On the Aishell-1 dataset, the experimental results of baseline and the ISPD-TDNN and JLF-ISPD-TDNN models proposed in this paper are shown in Table 2. The experimental results show that the JLF-ISPD-TDNN model and ISPD-TDNN had a better performance improvement in EER and minDCF without changing the parameters.

Table 2. Results on the Aishell-1 test set. The winner is in **bold**.

Model	Parameter (M)	EER (%)	minDCF
D-TDNN	2.82	2.45	0.2617
MFA-Conformer	20.8	1.70	0.2033
SE-ResNet	23.6	1.51	0.1308
SPD-TDNN	3.11	1.23	0.1432
ISPD-TDNN	3.11	1.19	0.1272
JLF-ISPD-TDNN	3.11	**1.02**	**0.1221**

Specifically, compared with the baseline SPD-TDNN model, the ISPD-TDNN model did not increase the number of parameters. Its EER was improved by 3.36%, and its minDCF was improved by 12.6%. The result indicates that when we adjust the position of the BN layer, the model can better learn the feature representation of data, enhance its nonlinear ability, and improve its stability and generalization ability. Compared to the baseline SPD-TDNN model, the JLF-ISPD-TDNN model achieved a relative improvement of 20.6% in EER and 17.3% in minDCF. Compared with ISPD-TDNN, JLF-ISPD-TDNN had a relative increase of 16.7% in EER and a relative increase of 4.2% in minDCF, which indicates that the joint loss function can make the neural network learn better speech representation, thereby improving the performance of speaker recognition. Compared to the baseline paper's D-TDNN model, our proposed model only has a slight parameter increase of 0.29, yet it achieves an inference accuracy of more than double. This achievement is attributed to both the exceptional context information extraction capability of the foundational SPD-TDNN model and the benefits brought by the joint loss function.

Compared to MFA-Conformer and SE-ResNet, JLF-ISPD-TDNN exhibits a significant improvement in performance. This is due to the fact that the variants of the transformer (conformer) used in MFA-Conformer struggle to effectively capture global contextual information. On the other hand, SE-ResNet's limited receptive field hinders it from adequately capturing local information. These shortcomings in both models result in embeddings lacking the necessary features, rendering them incapable of robust speaker identification.

5. Conclusions

To further improve the performance of speaker recognition algorithms, we proposed a novel model called JLF-ISPD-TDNN. By adjusting the BN position, the ISPD-TDNN model reduces the influence of the BN layer on the model gradient and enhanced the model's generalization ability. In addition, the joint loss function can use the contrast learning strategy to make the model learn more robust speaker feature representation. The proposed JLF-ISPD-TDNN model not only fully utilizes the ability of SPD-TDNN to better learn global contextual information, but also enables the network to better learn the similarities and differences between speakers. Therefore, this framework can be used to obtain more accurate speaker embedding representation.

Author Contributions: Conceptualization, methodology, software, validation, data curation, writing, original draft preparation, visualization, T.F.; formal analysis, investigation, resources, T.F., H.F. and S.C.; writing—review and editing, T.F., H.F., F.G., S.C. and C.L.; supervision, project administration, funding acquisition, F.G. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (12204062) and the Shandong Provincial Natural Science Foundation (ZR2022MF330).

Data Availability Statement: Not applicable.

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

References

1. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker Verification Using Adapted Gaussian Mixture Models. *Digit. Signal Process* **2000**, *10*, 19–41. [CrossRef]
2. Dehak, N.; Kenny, P.; Dehak, R.; Dumouchel, P.; Ouellet, P. Front-End Factor Analysis for Speaker Verification. *IEEE Trans. Speech Audio Process.* **2011**, *19*, 788–798. [CrossRef]
3. 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. [CrossRef]
4. Variani, E.; Lei, X.; McDermott, E.; Lopez-Moreno, I.; Gonzalez-Dominguez, J. Deep neural networks for small footprint text-dependent speaker verification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, 4–9 May 2014; pp. 4052–4056.
5. Snyder, D.; Garcia-Romero, D.; Povey, D.; Khudanpur, S. Deep Neural Network Embeddings for Text-Independent Speaker Verification. In Proceedings of the Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; pp. 999–1003.
6. 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 2018, Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
7. Desplanques, B.; Thienpondt, J.; Demuynck, K. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 3830–3834.
8. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
9. Gao, S.; Cheng, M.; Zhao, K.; Zhang, X.; Yang, M.; Torr, P.H.S. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef] [PubMed]
10. Lee, J.; Nam, J. Multi-Level and Multi-Scale Feature Aggregation Using Sample-level Deep Convolutional Neural Networks for Music Classification. *CoRR* **2017**, arXiv:1706.06810.
11. Gao, Z.; Song, Y.; McLoughlin, I.; Li, P.; Jiang, Y.; Dai, L. Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 361–365.
12. Yu, Y.; Li, W. Densely Connected Time Delay Neural Network for Speaker Verification. In Proceedings of the Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Shanghai, China, 25–29 October 2020; pp. 921–925.
13. Wan, Z.; Ren, Q.; Qin, Y.; Mao, Q. Statistical Pyramid Dense Time Delay Neural Network for Speaker Verification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Singapore, 23–27 May 2022; pp. 7532–7536.
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239.
15. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [CrossRef]
16. Wang, J.; Li, L.; Wang, D.; Zheng, T.F. Research on generalization property of time-varying Fbank-weighted MFCC for i-vector based speaker verification. In Proceedings of the 9th International Symposium on Chinese Spoken Language Processing, Singapore, 12–14 September 2014; p. 423.
17. Zhang, Y.; Lv, Z.; Wu, H.; Zhang, S.; Hu, P.; Wu, Z.; Lee, H.; Meng, H. MFA-Conformer: Multi-Scale Feature Aggregation Conformer for Automatic Speaker Verification. In Proceedings of the Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Republic of Korea, 18–22 September 2022; pp. 306–310. [CrossRef]
18. Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M.L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 5220–5224. [CrossRef]
19. Pearce, D.J.B.; Hirsch, H.G. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In Proceedings of the Interspeech, Beijing, China, 16–20 October 2000.
20. Snyder, D.; Chen, G.; Povey, D. MUSAN: A Music, Speech, and Noise Corpus. *CoRR* **2015**, arXiv:1510.08484.
21. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15–19 September 2019; pp. 2613–2617.

22. Deng, J.; Guo, J.; Yang, J.; Xue, N.; Kotsia, I.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5962–5979. [CrossRef] [PubMed]
23. Xiang, X.; Wang, S.; Huang, H.; Qian, Y.; Yu, K. Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition. *arXiv* **2019**, arXiv:1906.07317.
24. Le-Khac, P.H.; Healy, G.; Smeaton, A.F. Contrastive Representation Learning: A Framework and Review. *IEEE Access* **2020**, *8*, 193907–193934. [CrossRef]
25. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2019**, arXiv:1807.03748.
26. Gutmann, M.; Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; Volume 9, pp. 297–304.
27. Bu, H.; Du, J.; Na, X.; Wu, B.; Zheng, H. AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline. In Proceedings of the 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment, O-COCOSDA 2017, Seoul, Republic of Korea, 1–3 November 2017; pp. 1–5.
28. Snyder, D.; Garcia-Romero, D.; Sell, G.; McCree, A.; Povey, D.; Khudanpur, S. Speaker Recognition for Multi-speaker Conversations Using X-vectors. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, UK, 12–17 May 2019; pp. 5796–5800.
29. Wu, Z.; Yamagishi, J.; Kinnunen, T.; Hanilçi, C.; Sahidullah, M.; Sizov, A.; Evans, N.; Todisco, M.; Delgado, H. ASVspoof: The Automatic Speaker Verification Spoofing and Countermeasures Challenge. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 588–604. [CrossRef]