Cross-Scene Speaker Verification Based on Dynamic Convolution for the CNSRC 2022 Challenge

Jialin Zhang, Qinghua Ren, Youcai Qin, Zikai Wan, Qirong Mao*

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China

*Corresponding Author:mao_qr@ujs.edu.cn

Abstract

This paper mainly presents our developed approach for the CNSRC2022 competition, specifically the open and fixed tracks in speaker verification task. In the context of speaker verification, a standard protocol is to extract the discriminative feature embeddings to determine the speaker identity via the similarity calculation. Compared to the Voxceleb datasets, the Cnceleb datasets involve more complex conditions as well as more challenging scenarios, which increases multi-genre and cross-genre complexity greatly. For fixed track, we have proposed two main improvement options. In terms of the model architecture, adaptive convolution extracts more robust representations, while dynamic convolution improves the representation capacity of the model. In terms of the task, we find that the noisy scene information could bring the negative effect. To handle this problem, we adopt a gradient reversal layer to decouple the harmful scene features. For open track, we use a pre-trained model trained on the Voxceleb datasets, and then fine-tune it on the Cnceleb datasets. Finally, by fusing the scores of each system, our method achieves 0.4195 minDCF in the fixed track and 0.3707 minDCF in the open track.

1. Introduction

Speaker verification systems are becoming more and more popular in real-life applications, but they also face a variety of challenges. For example, noise interference, far-field speaker verification and disruption from scene information. As the research goes deeper [1, 2, 3], the framework of speaker verification systems is becoming clear. The main approach is to extract the feature embeddings of enrolling audio and the test audio by the trained model, then compare the similarity between them. Common input features are Mel Frequency Cepstrum Coefficient (MFCC), Filter bank (FBank), and even raw audio. Similarity calculation methods commonly used are Probabilistic Linear Discriminant Analysis (PLDA) as well as cosine similarity.

In the beginning, speaker verification has gone through traditional statistical methods such as Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), Gaussian Mixture model-Universal background model (GMM-UBM), Joint Factor Analysis (JFA), and i-vector. In recent years, speaker verification has been changing in terms of models as deep networks continue to develop. David Snyder et al. [4] has proposed x-vector model to extract speaker embeddings which gets perfect effect in 2017. Then in 2018, they complemented the experiment for exploring the effect of data augmentation on the results [5]. Since then, speaker verification methods using deep learning have become popular. Meanwhile, many variants of Time Delay Neural Network (TDNN) [6] have gradually emerged, including extended TDNN (E-TDNN) [7] and factorized TDNN (F-TDNN) [8]. Yaqi Yu et al. [9] proposed densely connected TDNN (D-TDNN) with bottleneck layers and dense connectivity. In 2020, Brecht Desplanques et al. proposed ECAPA-TDNN [10] using a variety of techniques in the field of vision, such as Squeeze-and-Excitation blocks. These complementary messages were also considered, aggregating and spreading the information of the different layers. In recent years, a number of new ideas and approaches have emerged in the field of speaker verification. A novel bidirectional multiscale feature aggregation (BMFA) [11] network is proposed to enable the repeated integration of features at different stages. A novel multi-scale waveform encoder [12] that tries to extract speaker embedding directly from the original waveform shows positive and competitive results. As in the case of text-independent speaker verification, text-dependent speaker verification has also been developed. Yan Liu et al. [13] improved text-dependent speaker verification using a multi-task learning network.

The Cnceleb1 [14] datasets and the Cnceleb2 [15] datasets used in this competition are collected in unconstrained conditions. Comparing to Voxceleb [16] datasets, Cnceleb datasets pay more attention to Chinese celebrities and scene genres of speech. This also leads to a more challenging problem. When a speaker's genre of enroll speech is different from the test speech, it can significantly reduce the accuracy of speaker verification.

All of our experiments were conducted based on the speechbrain [17] framework. For fixed track, to reduce the interference of scene genres of speech on speaker embedding, a variety of measures were considered. We tried various data processing, different model structures, and special ways of score normalization. For input samples, we determine the number of times to input based on the length of the utterance. The speech is randomly intercepted according to the specified duration. This can significantly increase the number of times the utterance is trained. The longer the utterance, the more times the speech is involved in training. Our main contribution in this track is the creative proposal of Dynet-ECAPA-TDNN and the use of gradient reversal layer to reduce the interference of scene information mismatch.

For open track, we tried three models, including ECAPA-TDNN model with 512 channels, ECAPA-TDNN model with 1024 channels, and multi-branch feature aggregation method based on multiple weighting (MBFG-MW), which adaptively learns attention weights for each branch to extract discriminative information that is beneficial to speaker verification. MBFA-MW is the model we proposed previously. The three models were first pre-trained on the Voxceleb datasets and then fine-tuned on the Cnceleb datasets.

For the final score files submitted to fixed and open tracks, we both performed a score fusion. The scores of each system

were first normalized. Then we considered fusion with average.

The rest of this paper is organized as follows. The section 2 mainly introduces the related work. Section 3 mainly describes the methods and techniques we used in this competition. Section 4 is the training protocol. The results of the experiments are placed in section 5. There are some discussions about the results in section 6. Finally, we draw some conclusions about this paper and discuss some future prospects in section 7.

2. Related Work

2.1. ECAPA-TDNN

ECAPA-TDNN has been able to get good results in recent years in all kinds of speaker verification competitions. So this time we firstly try to run the model on the Cnceleb datasets. Then make appropriate changes on top of that. ECAPA-TDNN contains a Res2Net [18] module which could get information through multi-scale receptive fields. ECAPA-TDNN also uses the Squeeze-and-Excitation module [19] in the field of vision that realizes information interaction between channels. Res2Net and Squeeze-and-Excitation module together form a SERes2Net block. In this competition, we tried ECAPA-TDNN models with input channels of 512 and 1024.

2.2. Dynamic Convolution

It is well known that the convolution parameters are shared for all samples. Dynamic convolution, however, can adaptively adjust the convolution parameters depending on the input samples. Some papers [20, 21, 22, 23, 24] have made developments and contributions to dynamic convolution. We applied dynamic convolution to ECAPA-TDNN in this competition and tried models with different number of convolution kernels to get good results by improving the expression of convolution.

2.3. Gradient Reversal Layer

Although the effectiveness of speaker verification continues to improve with the emergence of various new models and techniques, domain mismatch remains a current challenge in the field. A number of feature decoupling methods have also emerged to address this problem [25, 26]. The gradient reversal layer is used to perform data domain adaptation [27, 28]. Inspired by these methods, we come up with the idea of decoupling or fusion [29] the speaker embedding. By fusing scene information and speaker information, we can find that scene information has a very bad effect on the results. The scores of some models also show that the models do not have a good recognition rate for the genres of singing, movie, and drama. To eliminate the interference of scene information, we tried to adopt auxiliary adversarial tasks to learn scene-invariant speaker representations, connecting the speaker representation extraction module and the scene classification module through a gradient reversal layer.

3. Method

3.1. Dynet Block

For the Cnceleb datasets with complex scene information, we think that the model should have stronger representational power rather than a multi-scale receptive field.

In order to improve the expressibility of the model, we replaced the Res2Net module with the dynamic convolution module in SERes2Net block. We call this new block SE-DynetBlock. The dynamic convolution module includes an attention module to produce weights for the convolution layer. As far as we know, although dynamic convolution has been developed for a long time, this is the first time it has been applied in ECAPA-TDNN. The specific modifications can be seen in Figure 1, where the Res2Net module in the original model is replaced with a dynamic convolution module.

Dynamic convolution uses a set of K parallel convolution kernels $\{\tilde{W}_k, \tilde{b}_k\}$ instead of using one layer. Dynamic convolution can give the aggregated K sets of weights to the convolution layer. We have tried different values of K in our experiments. For each sample x of the input, the corresponding attention weights $\pi_k(x)$ are obtained by the attention mechanism. The final aggregation equation is as follows.

$$\tilde{W} = \sum_k \pi_k(x)\tilde{W}_k \tag{1}$$

$$\tilde{b} = \sum_k \pi_k(x)\tilde{b}_k \tag{2}$$

Dynamic convolution is a nonlinear function that has stronger representation capability compared to static convolution layers. At the same time, dynamic convolution is computationally efficient. This is because parallel convolution kernels share an output channel after aggregation. So it does not increase the depth or width of the model. All experiment results are presented in section 5.

3.2. Feature Decoupling

Although the idea of using auxiliary tasks to help with speaker verification come to mind. However, at first we are not sure whether feature fusion would be beneficial to the experimental results or feature decoupling would be effective to the experimental results. So we tried feature fusion without gradient reversal and also tried adding it for feature decoupling. We also want the modified model to adapt to the scene information domain. The use of the gradient reversal layer allows the gradient in scene classification to become the opposite number when backward.

Specific details of the model can be found in Figure 2(b). When we consider feature fusion, it is not necessary to add a gradient reversal layer to the auxiliary task. Only a classifier needs to be added after embedding to distinguish speaker scene information. The aim of this approach is to be able to distinguish both speaker and scene information through embedding for the purpose of feature fusion. The loss function of the model after adding the auxiliary tasks is as follows.

$$Loss1 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \tag{3}$$

$$Loss2 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y'_i}+m))}}{e^{s(cos(\theta_{y'_i}+m)} + \sum_{j=1,j\neq y'_i}^{n} e^{s\cdot cos\theta_j}} \tag{4}$$

$$Loss = Loss1 + Loss2 \tag{5}$$

where $Loss1$ denotes the loss in the speaker verification task and $Loss2$ denotes the loss in scene classification. Both of these loss functions use AAMsoftmax [30] loss. The speaker verification system has a total of c class labels, and y_i denotes the class of the i-th sample. Similarly, the auxiliary task has a total of n class labels, and y'_i denotes the class of the i-th sample. And N denotes the total number of samples, $cos\theta$ denotes the

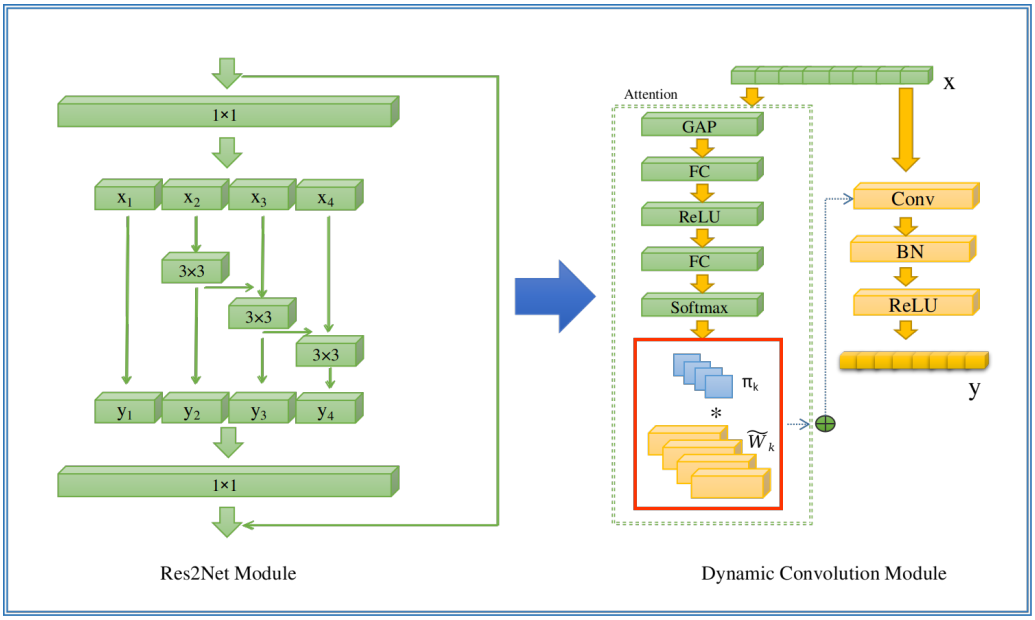


Figure 1: *Res2Net Module And Dynamic Convolution Module.* On the left, the Res2Net module in the original ECAPA-TDNN is replaced with the dynamic convolution module shown on the right.

angle between the weights and features, m denotes the penalty angle, and s is a hyperparameter that denotes the scaling factor.

When feature separation is considered, a gradient reversal layer is added. In this case, the gradient back propagation for update is shown below.

$$\theta_f = \theta_f - \mu(\frac{\partial Loss1}{\partial \theta_f} - \delta\frac{\partial Loss2}{\partial \theta_f}) \tag{6}$$

where δ is a hyperparameter that represents the coefficient when the loss of the auxiliary task is back-propagated. θ_f denotes the model gradient. And μ means learning rate.

4. Training Protocol

4.1. Datasets

Following with the competition rules, in the fixed track we only use the training and validation sets of the Cnceleb1 datasets and the Cnceleb2 datasets. The noise datasets RIRS_NOISE [31] are also used to enhance data. In the open track, we use the pre-training model trained on Voxceleb datasets. Then we fine-tune the model on Cnceleb datasets.

4.1.1. Cnceleb1 [14]

The Cnceleb1 datasets contain nearly 1000 speakers, 11 speech scenarios, and about 130,000 utterances. Most of the data are taken from the Bilibili platform. As we have done in the past, data below the specified duration are discarded. However, we do not recommend this at this time. Data with a duration of less than 2 seconds accounted for 32%, which is why we do not recommend discarding data less than 3 seconds. According to the evalution plan of the competition, a subset of 200 speakers selected from Cnceleb1 are used as the evaluation set on which the data scores of our entries were evaluated. Unlike the Voxceleb datasets, the Cnceleb datasets incorporate a manual checking step in the collection process. Therefore this datasets have fewer errors.

4.1.2. Cnceleb2 [15]

The Cnceleb2 datasets complement the Cnceleb1 datasets. There are more data, with nearly 2,000 speakers and 520,000 utterances. In addition to Bilibili, the data platform has added Changba, Himalaya, NetEase Cloud and Tiktok. It is worth noting that the dataset has not only speaker labels but also scene labels.

4.1.3. RIRS_NOISE [31]

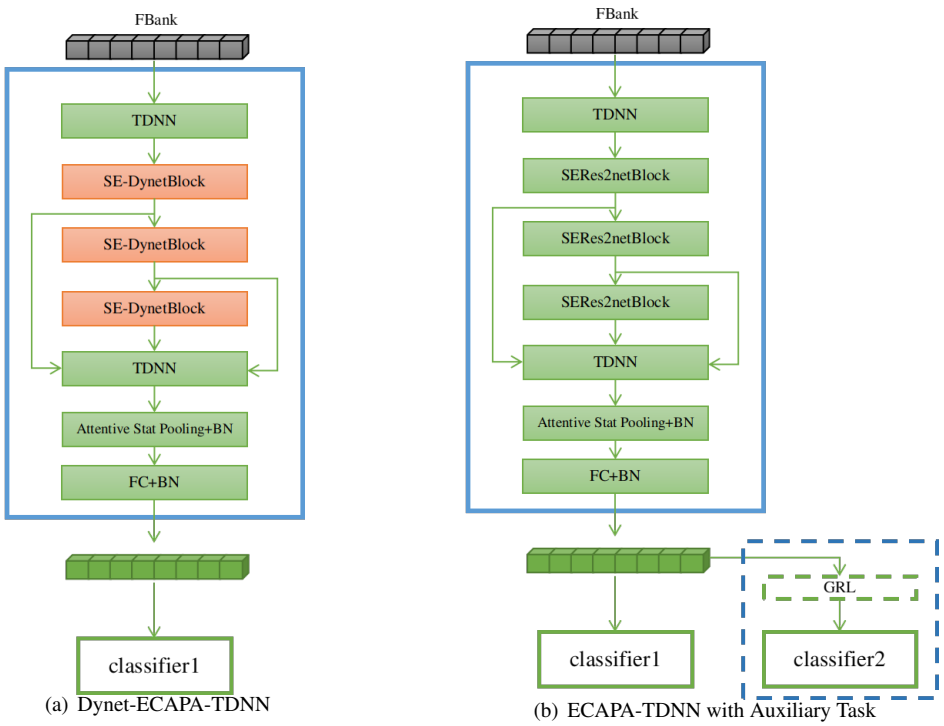
This database contains simulated and real indoor impulses and various noise data. It can be used to expand and enhance the datasets. The noise datasets are considered because the data in the Cnceleb datasets are closer to the real environment and the data contain more reverberation, noise, and music. Adding noise to the data can significantly improves the generalizability of the model.

4.1.4. Voxceleb [16]

There are datasets on speaker verification in the English language. We use this datasets on open track. Most of the data come from YouTube. The data are essentially gender-balanced (55% male). The celebrities had different accents, occupations, and ages. There were no overlap between the development and test sets. It is one of the more classic datasets in the field of speaker verification.

4.2. Data Process

We used λ seconds fixed data for training. Data longer than λ seconds are randomly intercepted. Data less than λ seconds are discarded. This will result in discarding too much data. And $60 * \lambda$ seconds of data is intercepted only once at random. This is not reasonable. We believe that a reasonable way to train is that the longer the data is, the more information it contains and the more times it takes part in the training. So we determine



(a) Dynet-ECAPA-TDNN (b) ECAPA-TDNN with Auxiliary Task

Figure 2: Architecture of the model.

the number of times the data is randomly intercepted according to the length of the data. All data longer than θ seconds are retained. And $60 * \theta$ seconds of data was intercepted 60 times at random. Each time, the interception is performed to the fixed length λ.

4.3. Data Augmentation

In general, the more data you have, the better trained the model is, and the more situations the data covers, the better the generalization and robustness of the model will be. Therefore, we explore various data augmentation methods, which we hope will alleviate the problem of dataset cross-scenario. Throughout the fixed and open track experiments, we set batch size to 16 or 32 and use the following data augmentation.

- **SpecAugment:** SpecAugment modifies the spectrogram by distorting the time domain signal, masking the frequency domain channel, and masking the time domain channel. This enhancement can be used to increase the robustness of the network to counteract distortions in the time domain and partial fragment losses in the frequency domain.
- **Speed peturbation [32]:** A slightly slower or faster signal is realized by resampling the audio signal at a rate similar to the original rate. Speech speed adjustment with 0.95 as well as 1.05 times.
- **Add reverse:** Only reverberations are added to the raw data.
- **Add noise:** Add only noise data to the original raw data. The range of the generated signal-to-noise ratio is 0-15.
- **Add reverse and noise:** Add both noise data and reverberation data to the original raw data.

The above data augmentation methods are performed at the same time. Therefore the data becomes six times of the original when training.

4.4. Features

There are many kinds of features of speech, such as MFCC, FBank. The response of the human ear to the sound spectrum is nonlinear. The FBank feature is a kind of feature similar to the one of human ear hearing. So in this experiment, we all use FBank as input feature. The length (in ms) of the sliding window used to compute the STFT is 25. The length (in ms) of the hop of the sliding window is 10. Number of samples to use in each stft is 400. Number of mel filters is 80. We finally obtained 80 dimensions of FBank features for training and testing.

4.5. Two Stages of Training

Inspired by the technique domain-based large margin fine-tuning [33], we also used this two-stage training approach. Our training solution is to start with θ=1,λ=2,for training. This has the advantage of retaining most of the data. If the duration of the speech does not reach λ, we will intercept the fragment of duration λ by speech concatenation afterwards. We call this phase the first stage. At this stage, the Adam optimizer with a weight decay of 2e-6 was used. The learning rate scheduling algorithm uses CyclicLR [34], with a maximum learning rate of 1e-3 and a base learning rate of 1e-8. The step size is set to half of the maximum number of iterations. We adopted AAMSoftmax loss with a scale of 30 and the margin is 0.2.

After training a few epochs, set θ to 3, λ to 6, and train a few more epochs. The intention is to use the primary data for fine-tuning. We call this phase the second phase. The maxi-

mum learning rate changed from 1e-3 to 1e-4 and the margin of AAMSoftmax changed from 0.2 to 0.4. Scale set to 60. With these two phases of training, we can usually get better results.

Due to the large amount of data and long training time, we found that most of the models can converge well in the second epoches. Therefore most of the models are only trained for two epoches.

4.6. Scores Normalization

Because the Cnceleb datasets are more close to the real environment, there are different scenario information. We use as-norm [35] for score normalization during testing in order to reduce the influence of various environmental differences between the test speech and the enrolled speech on the scores.

4.7. Fixed track

According to evaluation plan, the purpose of this track is to compare the effects of different algorithms on the same dataset. We firstly tried ECAPA-TDNN. Then we tried various dynamic convolution models and explored the effect of feature fusion and feature decoupling on the results.In the score fusion phase, we tried two approaches. One is to average the scores of multiple models, and the other is to use thresholds to determine the corresponding weights for each score. We first average the threshold values for each score file, and then each score is subtracted from the threshold and the absolute value is taken. The larger the absolute value, the more reliable this score is. The absolute values are then passed through softmax to obtain the corresponding coefficient weights. Then each score is linearly weighted and summed.

4.8. Open track

For open track, we can use the whatever data we want. The purpose of this track is to explore the current technology to achieve the best results on Cnceleb datasets. In this track, we additionally used the Voxceleb and noise datasets. As with the fixed track, we also consider the use of a two-stage training approach. Strategies such as score imputation and score fusion were also used.

5. Results

This section focuses on the results of various experiments. It is presented in two tracks. There are two main measures of the experimental results, which are minimum Detection Cost Function (minDCF) and Equal Error Rate (EER).

5.1. Fixed Track

In this track, we mainly use the training and validation sets of Cnceleb1 and 2. And we redivide the test and validation sets after merging them together. Add noise and reverberation using the RIRS_NOISE datasets. The training is divided into two stages.

The first stage uses data processed in the way θ=1,λ=2 and batch_size=32. The loss function uses an Additive Angular Margin softmax (AAMsoftmax) with the margin= 0.2 and scale s = 30 for cosine similarity.

And the second stage uses data processed in the way θ=3,λ=6 and batch size=16. Also the loss function margin is set to 0.4 and scale is set to 60. The models of the experiments are as follows:

- **ECAPA-TDNN**: ECAPA-TDNN model with 512 channels.
- **ECAPA-TDNN1024**: ECAPA-TDNN model with 1024 channels.
- **Dynet4-ECAPA-TDNN**: The Res2Net module in the original ECAPA-TDNN is replaced with the dynamic convolution module. And 4 means the K value of the dynamic convolution module.
- **Dynet16-ECAPA-TDNN**: The Res2Net module in the original ECAPA-TDNN is replaced with the dynamic convolution module. And the K value of the dynamic convolution module is 16.
- **Model+sn**: Use scores normalization when doing tests.

The results of the first stage of the experiments are presented in Table 1 and the second stage in Table 2.

Table 1: *The results of the first stage.*

model	min_DCF(0.01)	EER(%)
ECAPA-TDNN	0.5243	10.1915
Dynet4-ECAPA-TDNN	0.5132	9.9852
Dynet16-ECAPA-TDNN	0.5045	9.7549
ECAPA-TDNN1024	0.4891	9.3508
ECAPA-TDNN+sn	0.4955	9.6908
Dynet4-ECAPA-TDNN+sn	0.4885	9.5635
Dynet16-ECAPA-TDNN+sn	0.4845	9.3918
ECAPA-TDNN1024+sn	0.4761	8.9946

Table 2: *The results of the second stage.*

id	model	min_DCF(0.01)	EER(%)
A	ECAPA-TDNN+sn	0.4584	9.9859
B	Dynet4-ECAPA-TDNN+sn	0.4571	10.2715
C	Dynet16-ECAPA-TDNN+sn	0.4498	10.0366
D	ECAPA-TDNN1024+sn	0.4409	9.5522

We explored the effects of feature fusion and feature decoupling. The results of the experiments are placed in Table 3. Initially, we trained two epochs per model due to the long training time. The model is explained as follows.

- **ECAPA-TDNN+GRL+num**: ECAPA-TDNN model with 512 input channels. Gradient reversal layers were also used. When the coefficient of the gradient reversal layer is 1 (num = +1), it indicates feature fusion. When the coefficient is negative, it indicates feature decoupling.
- **ECAPA-TDNN+GRL+num+epoch3**: Due to the long training time, initially we only train 2 epochs per model. However, considering that the training difficulty will increase after adding auxiliary tasks, the number of training epochs should be increased appropriately. So we trained one more epoch.
- **Models+sn**: Use scores normalization when doing tests.
- **Models+second stage**: As in the previous experiments, we also tried to fine-tune the model on data of different time lengths.

Table 3: *The results of feature fusion and feature decoupling.*

id	model	min_DCF(0.01)	min_DCF(0.001)	EER(%)
1	ECAPA-TDNN+GRL+1	0.6107	0.7463	12.2811
2	ECAPA-TDNN	0.5243	0.6430	10.1915
3	ECAPA-TDNN+GRL-1	0.5250	0.6456	10.0929
4	ECAPA-TDNN+GRL-0.1	0.5244	0.6510	10.4646
5	ECAPA-TDNN+GRL-0.01	0.5168	0.6384	10.5284
6	ECAPA-TDNN+GRL-0.001	0.5112	0.6345	10.2224
7	ECAPA-TDNN+GRL-0.0001	0.5125	0.6391	10.0551
8	ECAPA-TDNN+GRL-0.001-epoch3	0.5003	0.6235	10.0062
9	ECAPA-TDNN+GRL-0.0001-epoch3	0.5059	0.6383	9.7690
10	ECAPA-TDNN+GRL-0.0001+sn	0.4892	0.6097	9.5101
11	ECAPA-TDNN+GRL-0.001-epoch3+sn	0.4839	0.6103	9.4360
12	ECAPA-TDNN+GRL-0.0001-epoch3+sn	0.4886	0.6156	9.3562
13	ECAPA-TDNN+GRL-0.001-epoch3+sn+second stage	0.4453	0.5645	9.7895
14	ECAPA-TDNN+GRL-0.0001-epoch3+sn+second stage	0.4473	0.5748	9.8113
	Fusion(13+14+C+D)	0.4227	0.5413	8.7524
	Fusion with threshold(13+14+A+B+C+D)	0.4195	null	8.8710

5.2. Open Track

In the open track, we first use Voxceleb datasets for pre-training and then fine-tune on Cnceleb datasets. Finally, a second fine-tuning is performed using data of different time lengths. We tried three models, including ECAPA-TDNN model with 512 input channels, ECAPA-TDNN model with 1024 input channels, and MBFA-MW. MBFA-MW is the model we proposed previously. The results of the open track are placed in Table4. After two stages of fine-tuning, a score fusion was performed and a final 0.3706 minDCF was achieved.

Table 4: *The results for open track.* Model 4-6 are second stage.

id	model	min_DCF(0.01)	EER(%)
1	MBFA-MW	0.4306	8.0651
2	ECAPA-TDNN	0.4395	9.1072
3	ECAPA-TDNN1024	0.4097	7.8969
4	MBFA-MW	0.3905	8.1779
5	ECAPA-TDNN	0.3986	9.0416
6	ECAPA-TDNN1024	0.3812	8.2634
	Fusion(4+5+6)	0.3706	7.3838

6. Discussion

In this section, we focus on the effect of the techniques used on the experiments. Through the experimental results, we can observe the improvement of each skill.

6.1. Effect of Dynamic Convolution

Both dynamic convolutions that we tried worked better than the baseline ECAPA-TDNN. The first three experiments in Table1 show that dynamic convolution has a certain enhancement on the experimental results, and the K parameter of dynamic convolution also has an effect on the experimental results. This proves that the CNSRC2022 task, based on complex scene information, requires models with more powerful characterization capabilities.

6.2. Effect of Gradient Reversal Layer

The results of the experiment are in Table3. **ECAPA-TDNN+GRL+1** denotes the experiment of feature fusion, where the coefficient of the gradient reversal layer is set to 1. It allows embedding to distinguish both the speaker and the scene. However, compared with the baseline ECAPA-TDNN, the test result changed from 0.52 to 0.61. This indicates that the scene information will interfere with the efficiency of speaker verification.

The coefficient of the gradient reversal layer is set to negative numbers in an attempt to perform feature decoupling. This allows embedding to distinguish only the speaker, but not the scene information. The experiments show that the coefficient keeps changing while the results keep improving. Eventually the results proved to be better when the coefficient is -0.001. Subsequently, it was considered that the addition of auxiliary tasks might increase the training difficulty. Therefore, one more training epoch is need.

In the end, the best result of the gradient reversal layer model reaches 0.5003. And 0.4453 is reached during the second stage of fine-tuning.

This shows that it is beneficial to remove the interference of scene information. The effect of the gradient reversal layer did not meet the expectation. We will continue to do some research in the future.

6.3. Effect of Scores Normalization

As can be seen in Table1, score normalization is a very effective technique. Each model can be improved by 0.03 after score normalization. Experimentally, score normalization is shown to be effective in reducing the effect of environmental differences between utterances.

7. Conclusion

In this paper, we present the models and techniques used by our team in the CNSRC2022 competition. We find that the scene information contained in the utterance interferes with the effect of speaker verification. We have two main contributions in this competition. One is a creative modification of ECAPA-TDNN including a dynamic convolution module to give the

model stronger representational capability. The second is the use of gradient reversal layer to decouple the features and reduce the interference of scene information. We finally achieve an 0.4195 minDCF in the fixed track and a 0.3707 minDCF in the open track.

Although feature decoupling using the gradient reversal layer get promoted, it did not achieve our expected results. Trying a more efficient way of feature decoupling is the next direction of our work.

8. Acknowledgements

This work is supported in part by the Key Projects of the National Natural Science Foundation of China under Grant U1836220, the National Nature Science Foundation of China under Grant 62176106. Jiangsu Province key research and development plan under Grant BE2020036.

9. References

[1] Daniel Garcia-Romero, Greg Sell, and Alan Mccree, "MagNetO: X-vector Magnitude Estimation Network plus Offset for Improved Speaker Recognition," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2020, pp. 1–8.

[2] Yingke Zhu and Brian Mak, "Orthogonality Regularizations for End-to-End Speaker Verification," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2020, pp. 17–23.

[3] Jee-Weon Jung, Ju-Ho Kim, Hye-Jin Shim, Seung bin Kim, and Ha-Jin Yu, "Selective Deep Speaker Embedding Enhancement for Speaker Verification," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2020, pp. 171–178.

[4] David Snyder, Daniel Garcia Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[5] David Snyder, Daniel Garcia Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[6] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 328–339, 1989.

[7] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2019, pp. 5796–5800.

[8] Jesús Villalba, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Fred Richardson, Suwon Shon, François Grondin, Réda Dehak, Leibny Paola García-Perera, Daniel Povey, Pedro A. Torres-Carrasquillo, Sanjeev Khudanpur, and Najim Dehak, "State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18," in *Interspeech*, 2019, pp. 1488–1492.

[9] Yaqi Yu and Wujun Li, "Densely connected time delay neural network for speaker verification," in *INTERSPEECH*, 2020.

[10] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *Interspeech*, 2020.

[11] Jiajun Qi, Wu Guo, and Bin Gu, "Bidirectional Multiscale Feature Aggregation for Speaker Verification," in *Proc. Interspeech*, 2021, pp. 71–75.

[12] Ge Zhu, Fei Jiang, and Zhiyao Duan, "Y-Vector: Multiscale Waveform Encoder for Speaker Embedding," in *Proc. Interspeech*, 2021, pp. 96–100.

[13] Yan Liu, Zheng Li, Lin Li, and Qingyang Hong, "Phoneme-Aware and Channel-Wise Attentive Learning for Text Dependent Speaker Verification," in *Proc. Interspeech*, 2021, pp. 101–105.

[14] Yue Fan, Jiawen Kang, Lantian Li, Kaicheng Li, Haolin Chen, Sitong Cheng, Pengyuan Zhang, Ziya Zhou, Yunqi Cai, and Dong Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7604–7608.

[15] Lantian Li, Ruiqi Liu, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, "Cn-celeb: Multi-genre speaker recognition," *Speech Commun.*, pp. 77–91, 2022.

[16] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.(CSL)*, 2020.

[17] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Juchieh Chou, SungLin Yeh, Szuwei Fu, Chienfeng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021.

[18] Shanghua Gao, Mingming Cheng, Kai Zhao, Xinyu Zhang, Minghsuan Yang, and Philip H. S. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.(TPAMI)*, pp. 652–662, 2021.

[19] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.(TPAMI)*, pp. 2011–2023, 2020.

[20] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam, "Soft conditional computation," *CoRR*, 2019.

[21] Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun, "Weightnet: Revisiting the design space of weight networks," in *Computer Vision (ECCV)*, 2020, pp. 776–792.

[22] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu, "Dynamic convolution: Attention over convolution kernels," *CoRR*, 2019.

[23] Yikang Zhang, Jian Zhang, Qiang Wang, and Zhao Zhong, "Dynet: Dynamic convolution for accelerating convolutional neural networks," *CoRR*, 2020.

[24] Chao Li, Aojun Zhou, and Anbang Yao, "Omni-dimensional dynamic convolution," in *International Conference on Learning Representations,(ICLR)*, 2022.

[25] Mufan Sang, Wei Xia, and John H.L. Hansen, "Deaan: Disentangled embedding and adversarial adaptation network for robust speaker representation learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6169–6173.

[26] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang, "Defrcn: Decoupled faster R-CNN for few-shot object detection," in *IEEE/CVF International Conference on Computer Vision, (ICCV)*, 2021, pp. 8661–8670.

[27] Yaroslav Ganin and Victor S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, 2015, pp. 1180–1189.

[28] Youcai Qin, Qirong Mao, Zhongchen Ma, and Jingjing Chen, "Learning device-invariant and location-invariant embedding for speaker verification using adversarial multi-task training," in *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*. IEEE, 2020, pp. 1–5.

[29] Qinghua Ren, Shijian Lu, Jinxia Zhang, and Renjie Hu, "Salient object detection by fusing local and global contexts," *IEEE Transactions on Multimedia*, pp. 1442–1453, 2021.

[30] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2019, pp. 4690–4699.

[31] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2017, pp. 5220–5224.

[32] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015, pp. 3586–3589.

[33] Miao Zhao, Yufeng Ma, Min Liu, and Minqiang Xu, "The speakin system for voxceleb speaker recognition challange 2021," *CoRR*, vol. abs/2109.01989, 2021.

[34] Leslie N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision, (WACV)*, 2017, pp. 464–472.

[35] Pavel Matejka, Ondrej Novotný, Oldrich Plchot, Lukás Burget, Mireia Díez Sánchez, and Jan Cernocký, "Analysis of score normalization in multilingual speaker recognition," in *Interspeech*, 2017, pp. 1567–1571.