NPU-HC Speaker Verification System for Far-field Speaker Verification Challenge 2022

Li Zhang[1], *Yue Li*[1], *Namin Wang*[2], *Jie Liu*[2], *Lei Xie*[1]

[1]Audio, Speech and Language Processing Group (ASLP@NPU), School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]Huawei Cloud

lizhang.aslp.npu@gmail.com, lxie@nwpu.edu.cn

Abstract

This report describes the NPU-HC system submitted to the Far-field Speaker Verification Challenge 2022 (FFSVC2022). In this challenge, the major problem is *domain mismatch* which lies between the enrollment and test utterances as well as the pre-train dataset (VoxCeleb) and the in-domain dataset (FFSVC). To mitigate this problem, we propose a two-stage transfer learning framework. Specifically, in the first stage, we adopt a speaker-aware weight-transfer method to fine-tune the pre-trained out-domain model with the FFSVC dataset and a part of the VoxCeleb dataset. The speaker-aware is obtained by evaluating the classification accuracy on the in-domain FFSVC data through the VoxCeleb pre-trained out-domain model, with the aim to maintain its strong speaker discrimination ability in the in-domain model. In the second stage, we use a speaker-center transfer learning method under a teacher-student framework to learn a domain-invariant embedding space. Specifically, the speaker embedding space of the near-field data trained teacher model guides the student model during its training with the FFSVC data. Moreover, we adopt the model soup strategy to average the weights of multiple models and use adaptive symmetrical score normalization (as-norm) in score fusion. Our approach leads to superior performance and comes to the second place in both challenge tracks.

Index Terms: far-field speaker verification, transfer learning, teacher-student model

1. Introduction

The far-field speaker verification challenge (FFSVC) series have particularly focused on the challenging far-field speaker verification (SV) task. Different from the previous challenge [1] that addresses multi-channel cross-domain SV, FFSVC2022 focuses on *single-channel* cross-domain SV [2], which means both the enrollment and test samples are single-channel data. Specifically, two challenge tasks are designed respectively to address fully-supervised (Task 1) and semi-supervised (Task 2) scenarios. The training datasets allowed to use include VoxCeleb 1&2 [3] and the FFSVC data [1], while the speaker labels of the FFSVC dataset are not allowed to use in Task 2 for semi-supervised learning purpose.

The major challenge of FFSVC2022 is the *two-fold* domain mismatch problem: 1) *training data mismatch* between the pre-train dataset (VoxCeleb) and the in-domain dataset (FFSVC) and 2) *enroll-test mismatch* at channel, time and text scales as well as same-gender difficult trails. To address the first mismatch, we propose a speaker-aware weight-transfer method to adapt the pre-trained VoxCeleb model to the FFSVC dataset. To tackle the second mismatch problem, we adopt a teacher-student framework to learn a domain-invariant speaker embedding space, where a speaker-center transfer loss is particularly introduced. Specifically in Task 2, we first use k-means clustering [4] to obtain pseudo labels, and then the above two-stage approach is naturally adopted.

2. Data Preparation

We use the following data to train our models:

- VoxCeleb 1&2 development sets
- FFSVC2020 training set, including its supplementary data
- FFSVC2020 development set

The development trials and test trials are released by the official organizers.

Online data augmentation has been successfully applied in various speech and speaker recognition tasks [5], leading to substantial performance gain. Thus in our approach, online data augmentation [6] is also adopted during the training of all our speaker embedding models, including the following aspects.

- **Frequency-domain SpecAug:** SpecAugment is applied directly to the log Mel-filter bank feature with masking blocks of both frequency channels and time steps [7].
- **Additive noise:** We add noise, music and babble from MUSAN [8] to the original waveform.
- **Reverberation:** We simulate reverberant speech by convolving clean speech with different RIRs from [9].
- **Time-domain wave masking:** We randomly mask some part of an audio waveform at time scale.
- **Speed perturb:** We adopt speed perturbation (0.9 and 1.1 times) to address the possible speed mismatch between enroll and test utterances.

3. Feature Extraction

In this challenge, the FFSVC data have three types of sample rates – 16kHz, 48kHz and 44.1kHz. Since the sampling rate of VoxCeleb is 16kHz, we first re-sample all the FFSVC samples to 16kHz. Then we extract 80-dimensional log Mel-filter bank energies from 16kHz raw input signals. The speaker embedding models are trained with log Mel-filter bank features with 25ms window size and 10ms window shift. Global mean and standard deviation (std) normalization is also applied.

4. Two-stage Transfer Learning

The proposed two-stage transfer learning approach is composed of a speaker-aware weight-transfer stage and a speaker-center transfer learning stage, as shown in Figure 1. Specifically in the first stage, we introduce a weight-transfer regularization loss to restraint the out-domain speaker embedding model without forgetting the discrimination ability of the pre-trained model. In details, the out-domain model is learned from the large but out-of-domain VoxCeleb datasets and we aim to reserve its strong speaker discrimination ability in the in-domain model which is obtained by fine-tuning the out-domain model with the small size in-domain FFSVC data. In the second stage, we improve the previous multi-level transfer learning approach [10] to centralize the speaker embeddings according to the speaker labels in the teacher model. Specifically, in the teacher-student learning framework, the teacher model is initialized by the in-domain model trained in the first stage and fine-tuned by the near-field (iPhone recorded) speech. The centralized speaker embeddings are more robust than the single speaker embedding and they have a good speaker discrimination ability on near-field data. We generate speaker-center embedding space by averaging the iPhone data obtained speaker embeddings according to the speaker labels. Then we use the speaker embedding space from the teacher model to guide the student model learning with the proposed speaker-center transfer loss.

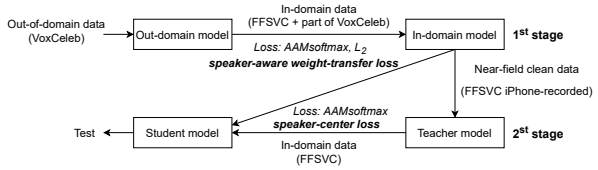

Figure 1: *The pipeline of the two-stage transfer learning approach with two particularly proposed losses – speaker-aware weight-transfer loss and speaker-center transfer loss – to address the domain mismatch problem.*

4.1. First Stage: Speaker-aware Weight Transfer

In the first stage, we first pre-train the speaker embedding model with VoxCeleb 1&2 development sets. This out-domain model has good speaker discrimination ability as it is trained using a large, noisy and heterogeneous dataset. Then we finetune the pre-trained out-domain model with the mixture data (FFSVC & a small part of VoxCeleb) with the hope to drag the model to the matched domain as well as to maintain its discrimination ability. To achieve this goal, in addition to AAMsotfmax for speaker classification, there are two additional weight-constrained loss functions are adopted. Specifically, the loss function during model fine-tuning is

$$L_{ft} = L_{CE} + \omega \cdot L_{wt} + L_2 \tag{1}$$

where L_{CE} is the speaker classification loss (AAMSoftmax), ω is the speaker-aware weights learned from the pre-trained model, L_{wt} is the weight-transfer loss and L_2 is the common L2 regularization loss.

The speaker-aware weights aim to indicate the degree of transferability of each convolution kernel parameter in each layer, calculated according to the following process. First, we fix the VoxCeleb pre-trained out-domain model and replace the last classification layer with a new layer whose number of nodes is equal to the number of speakers in FFSVC. Then we only train the new classification layer with FFSVC. We use this model to evaluate the effect of each convolution kernel parameter in each layer on speaker classification accuracy. Specifically, we mask the parameters of each convolution kernel in turn, and calculate the difference between the loss function after masking and the loss function before. This difference thus indicates the magnitude of the transferability of the in-domain FFSVC data for the convolution kernel. It is further normalized by softmax function, resulting in speaker-aware matrix ω.

Suppose W^0 is the parameters of the pre-trained speaker model, W^1 is the parameters of the fine-tuned speaker model. L_{wt} is calculated as

$$L_{wt} = \left\| W^0 - W^1 \right\|_2 . \tag{2}$$

The fine-tuned model is used as our in-domain model for the second stage transfer learning.

4.2. Second Stage: Speaker-center Transfer

In the second stage, we aim to alleviate the enroll-test mismatch, where the enrollment data is iPhone-recorded but test data can come from iPhone, iPad and microphone array. The in-domain model from the first stage is fine-tuned using the near-field iPhone-recorded data, resulting in the teacher model for second stage transfer learning. We first extract the speaker embeddings of iPhone recorded speech by the teacher model and then average all speaker embeddings according to the speaker labels, resulting in the speaker-center for each speaker. The speaker-center embedding space clearly depicts the relationship among different speakers as all speaker data are from the same device (iPhone) and relatively clean. This space is thus used as the teacher reference to guide the student model learning, where the target student model is to be trained using the FFSVC data. Specifically, we explore two forms of relationship learning. One is the MSE distance between near-field speaker-center embeddings and individual utterance embeddings extracted from the student model. Here the in-domain model from the first stage is used to initialize the student model. The other is angle-wise relationship measured by the near-field speaker-center embeddings and individual utterance embeddings extracted from the student model.

With a well trained teacher model, the embedding space generated by the teacher model has a more reliable reference compared with that of the student. Ideally, if there is no mismatch between teacher model and student model, they will have the same distribution in the embedding space. Suppose centralized embeddings extracted from teacher model t are $F_t(\theta_t) = [f_{\theta_t}(x_{t,1}), f_{\theta_t}(x_{t,2}), f_{\theta_t}(x_{t,3}), ..., f_{\theta_t}(x_{t,B})]$ with size $[B, F]$. The embeddings $F_s(\theta_s) = [f_{\theta_s}(x_{s,1}), f_{\theta_s}(x_{s,2}), f_{\theta_s}(x_{s,3}), ..., f_{\theta_s}(x_{s,B})]$ are extracted from the student model s. Here B denotes the training batch size and F is the dimension of embedding. Unlike previous work, we propose to preserve the pairwise instances distance calculated from $F_t(\theta_t)$ in $F_s(\theta_s)$. This solution aims to guide the student model towards the embedding space of the teacher model.

Specifically, the MSE distance between speaker embeddings from the teacher and student models is

$$L_M = \frac{1}{B^2} ||(f_{\theta_t} - f_{\theta_s})||_2, \tag{3}$$

where L_M is MSE distance of among speaker-center embeddings and speaker embeddings from the student model.

The angle-wise distance between speaker embeddings from the teacher and student models is defined as

$$L_A = \frac{1}{B^2}||(L_t - L_s)||_2, \tag{4}$$

where L_s and L_t are the cosine of the angle of any three speaker embeddings from teacher and student models. The formulas are as follows.

$$L_t = \cos\angle\left(\frac{f_{\theta_t}(x_{t,i}) - f_{\theta_t}(x_{t,j})}{\|f_{\theta_t}(x_{t,i}) - f_{\theta_t}(x_{t,j}))\|_2}, \frac{f_{\theta_t}(x_{t,i}) - f_{\theta_t}(x_{t,k})}{\|f_{\theta_t}(x_{t,i}) - f_{\theta_t}(x_{t,k}))\|_2}\right) \tag{5}$$

$$L_s = \cos\angle\left(\frac{f_{\theta_t}(x_{s,i}) - f_{\theta_t}(x_{s,j})}{\|f_{\theta_s}(x_{s,i}) - f_{\theta_s}(x_{s,j}))\|_2}, \frac{f_{\theta_s}(x_{s,i}) - f_{\theta_t}(x_{s,k})}{\|f_{\theta_s}(x_{s,i}) - f_{\theta_s}(x_{s,k}))\|_2}\right) \tag{6}$$

5. Model Fusion & Score Fusion

Model fusion, score normalization and fusion are adopted in our system with substantial performance gain.

5.1. Model Soup

Model soup is a more effective model fusion strategy [11] in which an ensemble of models is formed by averaging the weights of the models instead of combining each of their individual outputs. Thus in this challenge, we adopt the greedy soup method to improve the performance of model fusion. Specifically, for each type of neural model during training, we sort the models in decreasing order of minDCF on the development trials and choose the top five models for fusion. Then greedy soup is constructed by subsequently adding each model in the top five models as a potential ingredient in the soup, and we only keep the model in the soup if its performance on the development trials has improvement.

5.2. Score Normalization and Fusion

Score normalization aims to reduce within-trial variability for better calibration and more reliable threshold setting. In this challenge, we adapt symmetrical score normalization (s-norm) [12] to normalize the cosine scores of the test trials while the impostors are from the development trials.

Score fusion aims to further boost the performance by integrating multiple scores from different models which are expected to be complimentary. In the score fusion stage, we adopt manual calibration as well as automatic calibration. According to the performance on the FFSVC development set, we adopt the score level fusion that assigns different weights to different models. Considering that the model may over-fit on the development set with manual calibration, we particularly use the BOSARIS toolkit [13] for score calibration before score fusion.

6. Experiments

6.1. Model Structure

In this challenge, four models are trained with the following configurations.

- **ECAPA_TDNN (1024)** ECAPA-TDNN (1024) has 1024 channels in the frame-level convolution layers. The dimension of the bottleneck in the SE-Block and the attention module is set to 128. The scale dimension in the Res2Block is set to 8. The attention layer we use in ECAPA-TDNN (1024) is attentive statistic pooling (ASP) [14]. The embedding size in penultimate layer is 192. The parameters of this model is about 14 million.

Table 1: *Task 1 performance of the proposed approach on FFSVC 2022 development and evaluation sets. Here ECAPA-TDNN(1024) is used as the speaker embedding model.*

Model Index	Model Name	DEV		EVAL	
		EER(%)	minDCF(p=0.01)	EER(%)	minDCF(p=0.01)
A	ECAPA-TDNN (Out-domain)	8.462	0.714	-	-
A-1	+ finetune (In-domain)	6.662	0.593	-	-
A-2	+ weight transfer (In-domain)	5.811	0.516	6.211	0.565
A-3	+ speaker-aware weight-transfer (In-domain)	5.652	0.493	-	-
A-4	++ speaker-center loss (Student model)	4.674	0.411	-	-
A-5	+++ model soup	4.320	0.378	-	-
A-6	++++as-norm	3.921	0.356	4.033	0.359

- **ECAPA_TDNN (2048)** ECAPA-TDNN (2048) has 2048 channels in the frame-level convolution layers. The dimension of the bottleneck in the SE-Block and the attention module is set to 256. The scale dimension in the Res2Block is set to 8. The attention layer in ECAPA-TDNN (2048) is attentive statistic pooling (ASP) [14]. The embedding size in penultimate layer is 256. The parameters of this model is about 22 million.
- **ResNet34SE (256)** ResNet with squeeze and excitation attention (ResNet-SE) has achieved good performance in speaker verification [15, 16] recently. In ResNet34SE (256), we adopt ResNet34-SE with 128 channels of SE attention modules. The channel configuration of residual blocks is {32, 64, 128, 256}. The attention layer we use in ResNet34SE (256) self-attention pooling (SAP) [17]. The embedding size in penultimate layer is 256. The parameters of this model is about 12 million.
- **ResNet34SE (512)** In this model, we adopt ResNet34-SE with 128 channels of SE attention modules. The channel configuration of residual blocks is {64, 128, 256, 512}. The attention layer we use in ResNet34SE (512) self-attention pooling (SAP) [17]. The embedding size in penultimate layer is 512. The parameters of this model is about 25 million.

6.2. Experimental Setup

In this work, we adopt the additive angular margin loss (AAM-Softmax) [18] to train all models, where $s = 30$ and $m = 0.25$ are used for AAM-Softmax. The model training process is composed of base model training and fine-tuning. All model are first trained using the Adam optimizer [19] with a cyclical learning rate (CLR) using the triangular2 policy as described in [20]. The max and min learning rates are set at $1e-3$ and $1e-8$ respectively. The weight decay in the base model training stage is set to 2e-6. In the fine-tune stage, the max cyclical learning rate is reduced to $1e-4$ and the weight decay is $4e-4$. In the teacher-student model training stage, the max and min learning set at $1e-5$ and $1e-9$ respectively and the weight decay is $5e-4$. We use 16 pieces of NVIDIA V100 GPUs to train our models.

6.3. Experimental Results on Task 1

We first verify the effectiveness of the proposed approach using ECAPA-TDNN (1024) and results are summarized in Table 1. From the ablation in Table 1, we can see that all the tricks are effective with clear EER and minDCF reductions on the development trails. With all tricks, finally the best EER/minDCF on the development and evaluation trails are 3.921/0.356 and 4.033/0.359 respectively.

Then we apply the proposed approach (with all tricks)

to ECAPA-TDNN (2048), ResNet34SE-256 and ResNet34SE-512. Experimental results are shown in Table 2. We can see that the best single model is ECAPA-TDNN (2048) with 3.708%/0.339 in EER/minDCF on the evaluation set. After fusing the scores from ECAPA-TDNN (1024), ECAPA-TDNN (2048), ResNet34SE-256 and ResNet34SE-512, the final EER and minDCF are 3.470% and 0.319 on the evaluation set, which serves as the final score of our submitted system to Task 1.

Table 2: *Task 1 performance of the proposed approach on FFSVC 2022 development and evaluation sets. Here results on different models and model fusion are reported.*

Model Index	Model Name	DEV		EVAL	
		EER(%)	minDCF(p=0.01)	EER(%)	minDCF(p=0.01)
B	ECAPA-TDNN [2048]	3.673	0.342	3.708	0.339
C	ResNet34SE256 [256]	4.037	0.380	3.922	0.355
D	ResNet34SE512 [512]	3.530	0.349	3.662	0.340
Fusion	[A-6 & B & C & D]	-	-	3.470	0.319

6.4. Experimental Results on Task 2

In Task 2, we first use k-means [4] to generate pseudo speaker labels. Then we use the high-confidence utterances which are close to the cluster center to fine-tune the out-domain model. After getting the in-domain model, we select ten sentences of each speaker with the most accurate posterior probabilities from the in-domain model to generate the speaker-center embedding space. Finally, the speaker-center embedding space is used to guide the student model training with the FFSVC dataset.

Results on Task 2 are illustrated in Table 3. Finally, fusing scores from ECAPA-TDNN (2048) and ResNet34SE (512) leads to our best EER/minDCF of 5.342%/0.545, which is the final score of our system on Task 2.

Table 3: *Task 2 performance of the proposed approach on FFSVC 2022 development and evaluation sets.*

Model Index	Model Name	DEV		EVAL	
		EER(%)	minDCF(p=0.01)	EER(%)	minDCF(p=0.01)
E	ECAPA-TDNN [2048]	5.592	0.563	5.601	0.557
F	ResNet34SE512 [512]	5.611	0.576	5.842	0.566
Fusion	E &F	-	-	5.342	0.545

7. Conclusions

This report describes the NPU-HC team's system submitted to FFSVC2022. In this challenge, we have particularly proposed a two-stage transfer learning framework to deal with the domain mismatch problems. Specifically, speaker-aware weight-transfer is adopted to address the training data mismatch problem, while teacher-student learning framework with the proposed speaker-center transfer loss is adopted to address the enroll-test mismatch problem. Experiments show the effectiveness of the proposed two-stage transfer learning approach. Moreover, model soup fusion and adaptive symmetrical score normalization are also beneficial according to the experiments. With the above methods, the EER/minDCF scores of our system on the evaluation trials are 3.470%/0.319 and 5.342%/0.545 on Task 1 and Task 2 respectively.

8. References

[1] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The ffsvc 2020 evaluation plan," *Interspeech 2020 workshop*, 2020.

[2] X. Qin, M. Li, H. Bu, S. Narayanan, and H. Li, "Far-field speaker verification challenge (ffsvc) 2022: Challenge evaluation plan."

[3] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," *Interspeech*, 2017.

[4] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[5] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.

[6] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1038–1051, 2020.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech*, 2019.

[8] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[9] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep*, vol. 2, no. 2.4, p. 1, 2006.

[10] L. Zhang, Q. Wang, K. A. Lee, L. Xie, and H. Li, "Multi-level transfer learning from near-field to far-field speaker verification," *arXiv preprint arXiv:2106.09320*, 2021.

[11] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," *Proceedings of Machine Learning Research*, 2022.

[12] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *Interspeech*, 2017, pp. 1567–1571.

[13] N. Brümmer and E. De Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new dcf," *1304.2865*, 2013.

[14] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256.

[15] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *ICASSP 2021-2021*. IEEE, 2021, pp. 5814–5818.

[16] L. Zhang, Q. Wang, K. A. Lee, L. Xie, and H. Li, "Multi-Level Transfer Learning from Near-Field to Far-Field Speaker Verification," in *Proc. Interspeech 2021*, 2021, pp. 1094–1098.

[17] W. Cai, J. Chen, and M. Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, 2018, pp. 74–81.

[18] Y. Liu, L. He, and J. Liu, "Large Margin Softmax Loss for Speaker Verification," in *Interspeech*, 2019, pp. 2873–2877.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2014.

[20] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.