Augmentation Adversarial Training for Self-Supervised Speaker Representation Learning

Jingu Kang, Jaesung Huh, Hee Soo Heo, and Joon Son Chung

***Abstract*—The goal of this work is to train robust speaker recognition models using self-supervised representation learning. Recent works on self-supervised speaker representations are based on contrastive learning in which they encourage within-utterance embeddings to be similar and across-utterance embeddings to be dissimilar. However, since the within-utterance segments share the same acoustic characteristics, it is difficult to separate the speaker information from the channel information. To this end, we propose an augmentation adversarial training strategy that trains the network to be discriminative for the speaker information, while invariant to the augmentation applied. Since the augmentation simulates the acoustic characteristics, training the network to be invariant to augmentation also encourages the network to be invariant to the channel information in general. Extensive experiments on the VoxCeleb and VOiCES datasets show significant improvements over previous works using self-supervision, and the performance of our self-supervised models far exceeds that of humans. We also conduct semi-supervised learning experiments to show that augmentation adversarial training benefits performance in presence of speaker labels.**

***Index Terms*—Self-supervised learning, speaker recognition.**

I. Introduction

SPEAKER recognition is the ability to identify or verify a speaker's identity based on their voice. It has gained popularity in biometric authentication due to its easy accessibility and non-invasive nature.

Although there is a large body of recent literature on speaker recognition using deep neural network models [10], [20], [26], [46], [49], the overwhelming majority of these are based on the supervised learning framework. The availability of new large-scale datasets [12], [29], [32] combined with powerful neural network models has facilitated fast progress on many popular tasks within speaker recognition, but there are many challenges to extending this strategy to every application. For instance, the cost of annotating a new dataset can be prohibitively expensive and handling of sensitive biometric data can lead to privacy issues. The task of speaker verification is also very difficult for humans, resulting in inaccurate annotations in the absence of visual information.

On the other hand, there are many resources that can be used to learn representations, but have not been used due to the lack of annotations. For these reasons, unsupervised and self-supervised learning have recently received a growing amount of attention in order to leverage the abundant data available.

Existing literature on self-supervised learning of representations can be divided into two strands: *generative* or *discriminative*. Generative approaches learn representations by reconstructing the input data [22] or predicting withheld parts of the data, such as inpainting missing part of images [34] and colourising RGB images from only grey-scale images [50]. However, the element-wise generation is computationally expensive and is not necessary for representation learning.

Of relevance to our work is the second strand that learns discriminative representations directly, often using metric learning-based objectives. In particular, approaches based on contrastive learning in the latent space have shown to learn effective representations by taking within-class inputs from multiple views [4], [9], [30], [43] or modalities [3], [13], [14], [31], [39] of the same input data.

These strategies have been applied to speech signals in order to enable unsupervised learning of speaker representations. [36] samples two speech segments from same utterance and trains the network to maximise the mutual information between them. A key difference between supervised metric learning and the proposed contrastive learning framework is that segments from a single utterance have the same noise and reverberation characteristics. This effect has been partially mitigated using data augmentation in [23], which mimics the strategy of [9] that has shown promising performance in vision tasks.

A key challenge in speaker recognition is to learn embeddings that are speaker-discriminative, but invariant to all other spurious variations. Inspired by the work on domain adaptation using adversarial training [18], [44], recent works have used this framework to improve generalisation between languages [6], [7], [38] and between datasets [6], [48]. In particular, [10] and [27] have proposed channel-invariant training for speaker recognition by introducing a confusion loss between the same speaker segments from across and within an utterance.

While self-supervised experiments show promising results, the performance can be further improved by utilising small

Manuscript received 15 January 2022; revised 25 April 2022 and 3 August 2022; accepted 3 August 2022. Date of publication 22 August 2022; date of current version 14 October 2022. The work of Jaesung Huh was supported by Global Korea Scholarship. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Tara N. Sainath. *(Corresponding author: Joon Son Chung.)*

Jingu Kang is with the Inha University, Incheon 22212, South Korea (e-mail: jingu.zhenqiu.kang@gmail.com).

Jaesung Huh is with the Visual Geometry Group, University of Oxford, OX1 3PJ Oxford, U.K. (e-mail: jaesung@robots.ox.ac.uk).

Hee Soo Heo is with the Naver Corporation, Seongnam-si 13561, South Korea (e-mail: heesoo.heo@navercorp.com).

Joon Son Chung is with the Korea Advanced Institute of Science, and Technology, Daejeon 34141, South Korea (e-mail: joonson@kaist.ac.kr).

Digital Object Identifier 10.1109/JSTSP.2022.3200915

amounts of labelled data. We extend the work by fine-tuning the self-supervised models on small labelled datasets, and demonstrate the effectiveness of augmentation adversarial training also on semi-supervised learning. Finally, we compare the performance of the speaker verification models to humans, and show that our models significantly outperform humans.

II. Augmentation Adversarial Training

This section describes the proposed self-supervised training strategy. We describe the batch formation for training, then introduce the contrastive learning framework which samples two non-overlapping speech segments from each utterance and applies data augmentation. We then propose Augmentation Adversarial Training (AAT), which exploits an augmentation classifier in addition to speaker embedding extractor. Training is performed in turns to remove channel information from the speaker representation. We include the pseudo-code of AAT algorithm in the supplementary materials (Listing).

A. Batch Formation

Each mini-batch $\mathcal{B}$ contains randomly selected N utterances $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ out of set. For each utterance $\mathbf{x}_i$, we sample two non-overlapping speech segments, $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$, both of which are time-domain signals. Under the assumption that every utterance contains only one person's speech, $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ are from same identity.

B. Contrastive Training

Since $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ are sampled from the same utterance, the channel characteristics of the two segments are likely to be identical. As a result, using the standard metric learning methods, speaker embedding extractor might learn not only the speaker characteristics, but also the similarity of the environment between the two segments. Therefore, data augmentation such as additive noise or room impulse response (RIR) is applied to simulate different channel characteristics.

Specifically, for each two non-overlapping segments $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ $(1 \leq i \leq N)$, D-dimensional speaker embeddings $\mathbf{e}_{i,j,k}$ are computed as follows:

$$\mathbf{e}_{i,j,k} = f(\mathbf{x}_{i,j} * R_{i,k} + N_{i,k}) \quad (j,k) \in \{(1,1),(2,2)\} \tag{1}$$

where $R_{i,k}$ and $N_{i,k}$ are randomly selected from a set of pre-computed RIR filters and noise dataset, respectively. $f(\cdot)$ is the speaker embedding extractor and is trained with speaker loss functions. $*$ is the notation for convolution. Therefore, $\mathbf{e}_{i,j,k}$ refers to the embedding of j-th segment of i-th utterance, with augmentation type k.

Prototypical loss. Prototypical network [40] has been introduced for few-shot learning and has been shown to perform well in speaker verification [2], [11], [47]. In our case, $\mathbf{e}_{i,1,1}$ is a query and $\mathbf{e}_{i,2,2}$ is a prototype of size 1 support set. We compute the negative of the L2 distance as follows:

$$S(\mathbf{e}_i, \mathbf{e}_j) = -\|\mathbf{e}_i - \mathbf{e}_j\|_2^2 \tag{2}$$

In the angular variant of the prototypical loss (AP) [11], the distance function is replaced by a cosine similarity $sim(\cdot,\cdot)$ combined with learnable weight $w > 0$ and bias b:

$$S(\mathbf{e}_i, \mathbf{e}_j) = w \times sim(\mathbf{e}_i, \mathbf{e}_j) + b \tag{3}$$

where cosine similarity between $\mathbf{e}_i$ and $\mathbf{e}_j$ is defined as an inner product of normalised vectors:

$$sim(\mathbf{e}_i, \mathbf{e}_j) = \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{\|\mathbf{e}_i\|\|\mathbf{e}_j\|} \tag{4}$$

Cross entropy loss with a log-softmax function is used to minimise the distance between segments from same utterance and maximise the distance between different utterances.

$$L_{\text{spk}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(S(\mathbf{e}_{i,1,1}, \mathbf{e}_{i,2,2}))}{\sum_{i'=1}^{N} \exp(S(\mathbf{e}_{i,1,1}, \mathbf{e}_{i',2,2}))} \tag{5}$$

Contrast to supervised metric learning, it is not guaranteed that all $\mathbf{x}_i$ are from different speakers. If the batch size N is small relative to the total number of speakers and well-shuffled, it can be expected that most of the utterances in a batch are from different speakers.

C. Augmentation Adversarial Training

Data augmentation methods help the learnt embeddings to be more robust to channel variance, however do not explicitly remove the information from the embeddings. Since the augmentation methods simulate different channel environments, training the embeddings to be invariant to the augmentation also encourages the embeddings to be channel-invariant. Here, we propose Augmentation Adversarial Training (AAT) that penalises the ability to predict the augmentation in order to prevent the speaker embedding extractor from learning the channel information. The overview of this training method is in Fig. 1.

In addition to speaker representations $\mathbf{e}_{i,1,1}$ and $\mathbf{e}_{i,2,2}$, the third representation is extracted. The third representation $\mathbf{e}_{i,2,1}$ comes from the second segment $\mathbf{x}_{i,2}$. We apply same RIR filter $R_{i,1}$ and additive noise $N_{i,1}$ as $\mathbf{e}_{i,1,1}$, which is illustrated in left figure of Fig. 1.

$$\mathbf{e}_{i,j,k} = f(\mathbf{x}_{i,j} * R_{i,k} + N_{i,k}) \quad (j,k) \in \{(1,1),(2,1),(2,2)\} \tag{6}$$

Then, discriminator training phase and embedding training phase are performed alternately, as explained below.

Discriminator training. In this step, we train the augmentation classifier g. The assumption is that $\mathbf{e}_{i,1,1}$ and $\mathbf{e}_{i,2,1}$ share the same channel characteristic, while $\mathbf{e}_{i,1,1}$ and $\mathbf{e}_{i,2,2}$ have different characteristics. We generate two types of input per each mini-batch, $\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,1}$ and $\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,2}$, where $\mathbin{+\!\!+}$ indicates concatenation of vectors. Since $\mathbf{e}_{i,j,k}$ is D-dimensional vector, both inputs' dimensions are $2D$. The resultant batch size for training g is $2\,N$, N inputs of $\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,1}$ and another N inputs of $\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,2}$. The network is trained to classify whether two inputs are from the same channel by using binary cross entropy loss. In this step, the gradient does not flow to the speaker embedding extractor. The loss function L_{dis} can be formulated as below where $\sigma(\cdot)$ is a sigmoid function.

$$\begin{aligned} L_{\text{dis}} = -\frac{1}{2\,N}\sum_{i=1}^{N} &(\log(\sigma(g(\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,1}))) \\ &+ \log\,(1 - \sigma(g(\mathbf{e}_{i,1,1} \mathbin{+\!\!+} \mathbf{e}_{i,2,2})))) \end{aligned} \tag{7}$$

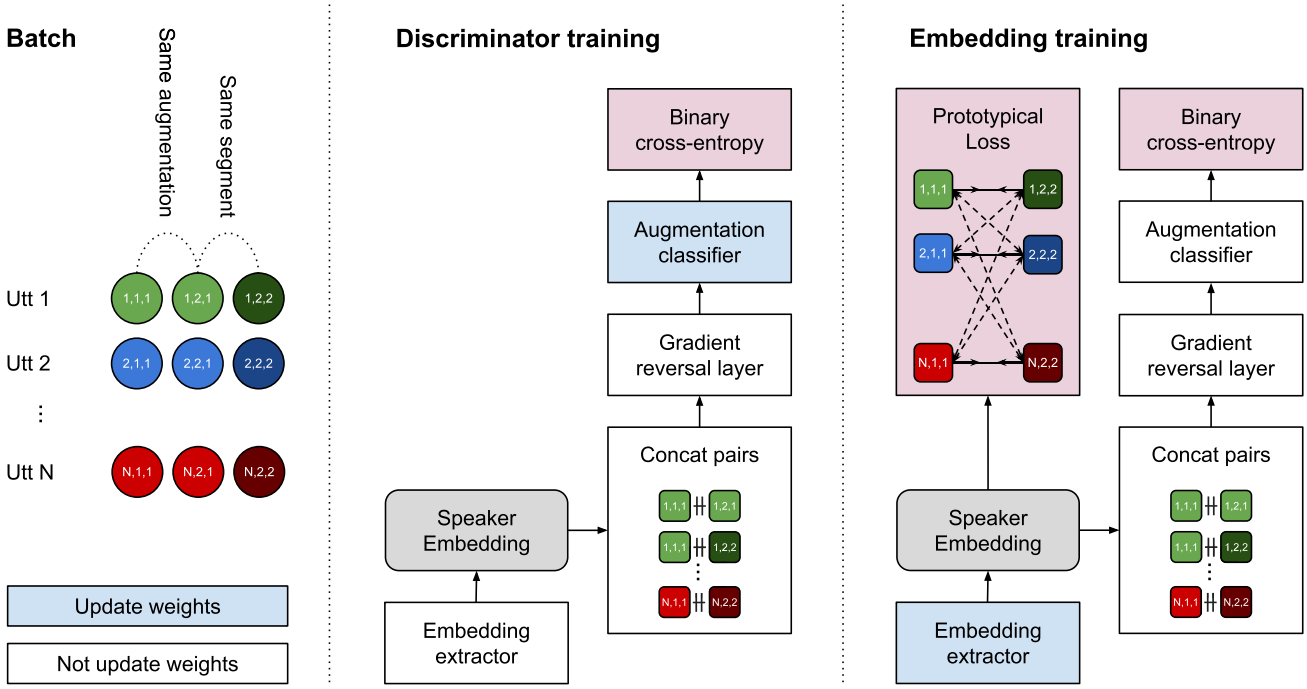


Fig. 1. Overview of the training strategy. The index notation for the inputs and the embeddings are consistent with the equations, i.e. i, j, k refer to j-th segment of i-th utterance, with augmentation type k. Best seen in colour.

Embedding training. In this step, we train the speaker embedding extractor f. While training f with $\mathbf{e}_{i,1,1}$ and $\mathbf{e}_{i,2,2}$ similar to Section II-B, we also apply Augmentation Adversarial Training loss (AAT loss) to encourage speaker embedding extractor to learn channel-invariant embedding. The weights of augmentation classifier g are *fixed* during this step. Learning objective related to this strategy is described below.

AAT loss. AAT loss is applied to remove the channel information from speaker embeddings. After training the augmentation classifier to distinguish channel similarities, we apply binary cross entropy loss which is similar to L_{dis}. One difference is, a gradient reversal layer is placed between embedding extractor and augmentation classifier, thereby penalising the ability to correctly predict whether the pair of segments share the same channel characteristics. It can be formulated as Equation 8 where minus indicates the use of gradient reversal layer.

$$L_{\text{aat}} = -L_{\text{dis}} \tag{8}$$

The overall loss is the summation of the speaker loss and the AAT loss with a weight parameter λ. L_{spk} can be either prototypical or angular prototypical loss function.

$$L_{\text{overall}} = L_{\text{spk}} + \lambda L_{\text{aat}} \tag{9}$$

D. Semi-Supervised Training

While there has been an increasing attention on self-supervised speaker representation, the best results reported using self-supervised methods fall far short of supervised counterparts. Semi-supervised learning methods utilise small amounts of labelled data to push the performance of self-supervised models closer to the fully supervised ones. For example in automatic speech recognition, pre-training using *wav2vec 2.0* [5] has demonstrated very strong performance while using orders of magnitude less labelled data than in previous works. Here, we apply semi-supervised learning strategy to speaker recognition.

Supervised fine-tuning. The self-supervised models trained in Sections II-B and II-C are fine-tuned with full supervision using a small amount of data. The experiments are initialised from the best models pre-trained using self-supervision. The fine-tuning is performed using a combination of softmax and angular prototypical losses, which has shown strong results in [25]. We compare pre-trained models trained with and without augmentation adversarial training to demonstrate its effectiveness.

III. Experiments

A. Input Representations

Since the utterances in VoxCeleb are always longer than 4 seconds, two 1.8-second segments are randomly sampled from each utterance during batch formation to construct two non-overlapping speech segments $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,2}$ introduced in Section II-A. The duration of the segments is slightly shorter than half of the shortest utterance in order to allow for small temporal perturbation. 40-dimensional log-mel spectrogram is extracted with window length 25 ms and hop length 10 ms. Instance normalisation [45] is performed as a mean variance normalisation to the input. We do not use voice activity detection (VAD) since the dataset mostly consists of continuous speech.

B. Network Architecture

The network architecture of the speaker embedding extractor closely follows the **Fast ResNet-34** [11] and **ECAPA-TDNN** [16] architectures. Fast ResNet-34 is a

lightweight version of the original ResNet-34 with the same architecture but the channel sizes are reduced to a quarter. Self-attentive pooling is performed on the output of residual blocks along the time axis, followed by a fully connected layer. The dimension of the speaker embedding $\mathbf{e}$ is 512.

ECAPA-TDNN is an advanced version of x-vector system [42] in the speaker recognition field. This architecture is composed of time-delay neural blocks (TDNNs) and squeeze and excitation (SE) layers unified with blocks of Res2Block layers. In addition, the model contains the layers for propagating and aggregating hierarchical features in one network. At the end of the network, the frame-level features are pooled by channel-dependent frame attention for the fixed dimensional embedding.

The **augmentation classifier** consists of a gradient reversal layer followed by two fully connected layers with hidden size 512. ReLU activation and one-dimensional batch normalisation layer are placed between these layers. The size of the last fully connected layer is 1 since the network is a binary classifier.

C. Data Augmentation

Data augmentation plays a crucial role in contrastive learning, as reported by previous literature in speaker recognition [23] and other domains [4], [9], [30], [43]. We exploit two popular augmentation methods in speech processing – additive noise and RIR simulation. For additive noise, we use the MUSAN corpus [41]; for room impulse responses, we use 1,000 pre-computed RIR filters. Both noise and RIR filters are randomly selected during training. The types of augmentation and the SNR range for each type are the same as those used by the original x-vector paper – see Section 3.3 of [42] for details.

In order to verify the effects of the different augmentation methods, we perform a number of experiments, (1) without any augmentation, (2) applying only noise addition, (3) applying *either* noise addition or reverberation and (4) applying *both* noise addition and reverberation. We also compare the results of only augmenting one of the speech segment (i.e. $\mathbf{e}_{i,1,1} = f(\mathbf{x}_{i,1})$, $\mathbf{e}_{i,2,1} = f(\mathbf{x}_{i,2})$, and $\mathbf{e}_{i,2,2} = f(\mathbf{x}_{1,2} * R_{1,2} + N_{1,2})$) and augmenting both of the speech segments.

D. Training Details

Our implementation is based on the PyTorch framework [33]. The models are trained using a NVIDIA V100 GPU with 32 GB memory. All experiments are repeated independently three times in order to minimise the effect of random initialisation. Mean and standard deviation of the results are reported in Section III-G.

Self-supervised training. We use the Adam optimiser with an initial learning rate of 0.001 decreasing by 5% every 5 epochs, for a total of 150 epochs. 200 utterances are randomly selected for each mini-batch formation.

Supervised fine-tuning. Our implementation follows the supervised training protocol in [11]. We use the Adam optimiser with an initial learning rate of 0.001 decreasing by 5% every 10 epochs, for a total of 500 epochs.

The fine-tuning experiments are initialised from the best performing self-supervised models with ECAPA-TDNN architecture trained using both noise and RIR augmentation.

E. Dataset

VoxCeleb. VoxCeleb is an audio-visual dataset consisting of short clips of human speech, extracted from celebrity interview videos uploaded to YouTube. The models are trained on the development set of VoxCeleb2 [12], which consists of over 1 million utterances from 5,994 speakers. Speaker labels in VoxCeleb2 are not used in our method. For the semi-supervised learning experiment, we also use the development set of VoxCeleb1 [32] dataset along with its labels, which consists of 148,642 utterances from 1,211 speakers. The original test set of VoxCeleb1 [32] containing 40 speakers is used for evaluation.

CN-Celeb. CN-Celeb [17] is a speaker recognition dataset consisting of speech data of Chinese celebrities, extracted from Bilibili using an automated pipeline similar to that used to collect VoxCeleb. The total set contains 130,109 utterances (274 hours) from 1,000 speakers, of which 800 speakers are designated for training and 200 for evaluation. The dataset is challenging since it contains different genres (entertainment, interview, singing, advertisements, etc.), most of the utterances involve strong real-world noise and many are short (32% of the utterances are less than 2 seconds). The dataset has been checked by humans.

VOiCES. The Voices Obscured in Complex Environmental Settings (VOiCES) [37] corpus contains speech recorded by far-field microphones in noisy room conditions. Evaluation on this dataset is performed to provide out-of-domain trial for the models trained on the VoxCeleb2 dataset. In particular, we use the evaluation list provided in the *development* data for the 2019 VOiCES challenge, which contains 4 million pairs from 15,904 utterances. Note that the speaker models are *not* trained or fine-tuned on this dataset, in order to verify that the models trained on the VoxCeleb dataset generalises to out-of-domain data.

F. Baselines

We compare the results of our methods with a range of baselines in Table I.

Previous works using self-supervision. Self-supervised methods have been used in speaker verification by a number of previous works. [31] uses *cross-modal* self-supervision to learn the joint representation of face images and speech segments. [14] extends this work by optimising within-modality distances as well as across-modality. [23] proposes audio-only self-supervised learning with data augmentation using additive noise and RIR filters, which is of closest relevance to our work. We denote the result of those works in Table I as **Disent.**, **CDDL** and **GCL**, respectively.

I-vectors. I-vectors [15] have been used widely in speaker recognition before the emergence of deep learning. Although the i-vectors are often used in conjunction with probabilistic linear discriminant analysis (PLDA) back-end to improve performance [8], [24], [28], training of i-vectors and scoring with cosine similarity as proposed by the original paper [15] do not require any supervision.

60-dimensional frame-level features (19 Mel-frequency cepstral coefficients + energy + Δ + $\Delta\Delta$) are extracted from the audio signal using a 25 ms window with 10 ms shifts, then mean and variance normalisation (MVN) is applied.

TABLE I
SPEAKER VERIFICATION PERFORMANCE ON THE VOXCELEB1 TEST SET

Objective	Model Aug.	ResNet34 EER (%)	ResNet34 MinDCF	ECAPA-TDNN EER (%)	ECAPA-TDNN MinDCF
Self-supervised baselines §					
Disent. [31]	-	22.09	-	-	-
CDDL [14]	-	17.52	-	-	-
GCL [23]	Noise or RIR	15.26	-	-	-
I-vector †	-	15.28	0.627	-	-
Human benchmark ‡§					
AMT	-	26.51	-	-	-
Expert	-	15.77	-	-	-
No augmentation					
P	-	27.30 ± 0.15	0.788 ± 0.002	-	-
AP	-	25.37 ± 0.15	0.788 ± 0.004	-	-
Augment one segment					
P	Noise	20.58 ± 0.30	0.738 ± 0.003	15.13 ± 0.00	0.650 ± 0.003
P + **AAT**	Noise	17.08 ± 0.55	0.685 ± 0.016	14.62 ± 0.21	0.627 ± 0.005
P	Noise or RIR	18.22 ± 0.42	0.719 ± 0.003	11.77 ± 0.37	0.585 ± 0.013
P + **AAT**	Noise or RIR	12.77 ± 0.60	0.634 ± 0.016	10.56 ± 0.16	0.546 ± 0.002
P	Noise and RIR	13.03 ± 0.05	0.610 ± 0.005	9.13 ± 0.03	0.487 ± 0.005
P + **AAT**	Noise and RIR	9.96 ± 0.33	0.522 ± 0.019	8.71 ± 0.22	0.458 ± 0.005
AP	Noise	18.63 ± 0.37	0.731 ± 0.004	13.03 ± 0.06	0.615 ± 0.001
AP + **AAT**	Noise	14.47 ± 0.06	0.666 ± 0.004	12.85 ± 0.04	0.597 ± 0.004
AP	Noise or RIR	16.43 ± 0.25	0.710 ± 0.006	10.44 ± 0.03	0.560 ± 0.007
AP + **AAT**	Noise or RIR	11.35 ± 0.18	0.612 ± 0.008	8.39 ± 0.08	0.471 ± 0.004
AP	Noise and RIR	11.43 ± 0.20	0.592 ± 0.013	8.08 ± 0.04	0.452 ± 0.014
AP + **AAT**	Noise and RIR	8.86 ± 0.18	0.490 ± 0.009	7.35 ± 0.12	0.387 ± 0.002
Augment both segments					
P	Noise	16.00 ± 0.05	0.667 ± 0.002	15.16 ± 0.07	0.637 ± 0.000
P + **AAT**	Noise	15.22 ± 0.24	0.640 ± 0.004	14.85 ± 0.27	0.634 ± 0.005
P	Noise or RIR	12.42 ± 0.15	0.623 ± 0.006	11.46 ± 0.18	0.573 ± 0.004
P + **AAT**	Noise or RIR	10.54 ± 0.06	0.544 ± 0.002	10.23 ± 0.14	0.526 ± 0.002
P	Noise and RIR	10.16 ± 0.16	0.524 ± 0.009	9.14 ± 0.16	0.477 ± 0.005
P + **AAT**	Noise and RIR	9.36 ± 0.07	0.482 ± 0.004	8.40 ± 0.23	0.447 ± 0.007
AP	Noise	14.73 ± 0.19	0.665 ± 0.006	13.08 ± 0.05	0.618 ± 0.003
AP + **AAT**	Noise	13.56 ± 0.18	0.632 ± 0.008	12.80 ± 0.19	0.597 ± 0.005
AP	Noise or RIR	11.60 ± 0.14	0.620 ± 0.004	10.51 ± 0.16	0.559 ± 0.006
AP + **AAT**	Noise or RIR	9.03 ± 0.07	0.512 ± 0.011	8.27 ± 0.22	0.454 ± 0.007
AP	Noise and RIR	9.56 ± 0.18	0.511 ± 0.011	8.03 ± 0.13	0.449 ± 0.010
AP + **AAT**	Noise and RIR	8.65 ± 0.14	0.469 ± 0.008	7.19 ± 0.06	0.386 ± 0.004

† Uses the I-vector together with cosine similarity. ‡ computed on a subset of 2,000 pairs. § is unrelated to network architecture. **P:** prototypical loss, **AP:** angular prototypical loss, **AAT:** augmentation adversarial training.

A gender-independent universal background model, containing 2,048 Gaussian components, and a total variability matrix with dimensionality 400 are trained, both with 10 iterations. Our implementation is based on the popular Kaldi [35] toolkit.

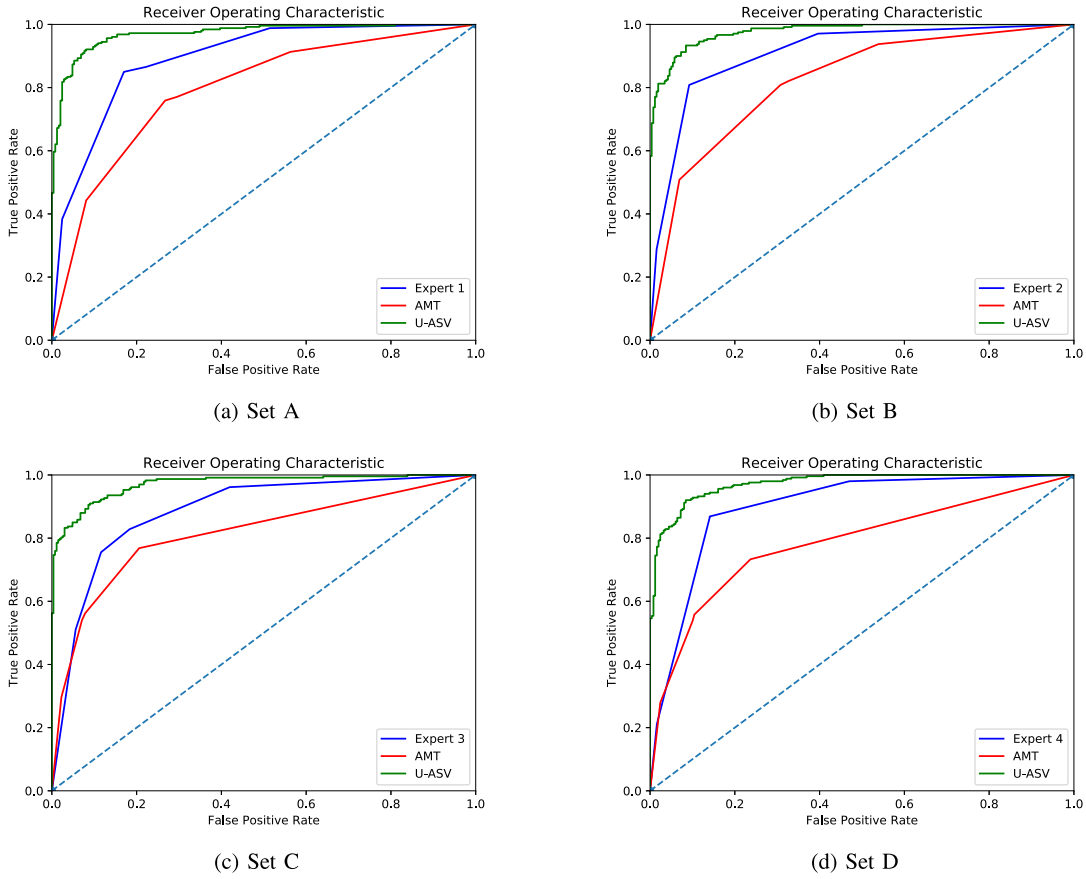
G. Human Benchmark

Humans do not learn how to recognise the speaker identity through supervised training as computers do. Therefore, it is interesting to compare the human performance on speaker verification as a self-supervised counterpart of our model. We conduct experiments with two groups of annotators – crowdworkers on Amazon Mechanical Turk (**AMT**) and experts who have dealt with speaker recognition for several years (**Experts**).

They are asked to annotate random subsets of the VoxCeleb test set. The evaluation protocols for these experiments mimic the VoxCeleb evaluation for automatic speaker recognition – the annotators are given utterance pairs, and they are asked whether they believe that the two utterances are spoken by the same speaker.

The annotators are given a pair of utterances to listen to, and are asked to choose between one of the following options. The annotators are discouraged from using the score of 3 (borderline). They are given up to 30 seconds for the task.

1 - Definitely different,
2 - Probably different,
3 - Borderline,
4 - Probably the same,
5 - Definitely the same.



(a) Set A (b) Set B

(c) Set C (d) Set D

Fig. 2. Receiver Operating Characteristic curves for the different subsets of the VoxCeleb1 test set. Set A, B, C and D are annotated by different experts. **U-ASV** represents the self-supervised ResNet model.

AMT. Amazon Mechanical Turk is a crowdsourcing marketplace to hire remotely located *crowdworkers* to perform discrete microtasks such as data annotation or surveys.

2,000 randomly sampled pairs from the VoxCeleb test set are given to the annotators through this platform, who are rewarded on a per-sample basis. The tasks are only made available to the most experienced and highly rated workers, however the annotators do not necessarily have previous experience in speaker recognition.

The annotators are told that approximately *half* of the pairs are from the speaker, and are given some example pairs to listen to before working on the task.

Experts. The samples are also annotated by the authors of this paper, who have several years of experience in speaker recognition. The authors are very familiar with the VoxCeleb dataset, including the statistics of the test set.

The same 2,000 pairs used by the Mechanical Turk are divided into 4 subsets of 500, each of which is annotated by a different author. These subsets are referred to as Sets A, B, C and D in Table VII.

IV. Results

Evaluation protocol. We report two performance metrics: (i) the Equal Error Rate (EER) which is the rate at which both acceptance and rejection errors are equal; and (ii) the minimum detection cost of the function (MinDCF) used by the NIST SRE [1] and the VoxSRC[1] evaluations. For computing EER, we sample 10 segments for each utterance and compute the mean of $10 \times 10 = 100$ distances from all possible combinations per trial pair in the evaluation set. This protocol is in line with that used by [10], [11]. The parameters $C_{miss} = 1$, $C_{fa} = 1$ and $P_{target} = 0.05$ are used for the minimum detection cost function. Please refer to [1] for the exact equation.

A. Self-Supervised Training

Performance improvement with AAT. Table I and Table II report the experimental results in VoxCeleb1 and VOiCES test sets, respectively. Data augmentation is a key to the performance of self-supervised speaker models. More aggressive augmentation schemes (e.g. noise and RIR) improve the performance of the models. This implies that data augmentation helps to train the noise-robust network and is essential to apply diverse channel effects.

In Table I, AAT reduces the verification errors across a range of augmentation settings, objective functions, and network architectures. The best performing model trained with angular prototypical loss and AAT achieves an equal error rate

[1][Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/voxceleb/competition2020.html

TABLE II
SPEAKER VERIFICATION PERFORMANCE ON THE VOICES TEST SET. SYMBOLS AND ACRONYMS ARE THE SAME AS THAT IN TABLE I

Objective	Model Aug.	ResNet34 EER (%)	ResNet34 MinDCF	ECAPA-TDNN EER (%)	ECAPA-TDNN MinDCF
Self-supervised baselines §					
I-vector †	-	17.49	0.817	-	-
No augmentation					
P	-	29.69 ± 1.45	0.992 ± 0.004	-	-
AP	-	32.21 ± 0.89	0.994 ± 0.002	-	-
Augment one segment					
P	Noise	22.04 ± 0.53	0.944 ± 0.002	16.90 ± 0.06	0.883 ± 0.007
P + **AAT**	Noise	18.98 ± 0.33	0.913 ± 0.012	17.42 ± 1.22	0.881 ± 0.008
P	Noise or RIR	17.27 ± 0.39	0.894 ± 0.006	8.92 ± 0.29	0.596 ± 0.003
P + **AAT**	Noise or RIR	12.96 ± 0.43	0.760 ± 0.011	7.74 ± 0.34	0.507 ± 0.023
P	Noise and RIR	11.94 ± 0.01	0.713 ± 0.012	5.11 ± 0.04	0.358 ± 0.005
P + **AAT**	Noise and RIR	9.05 ± 0.96	0.583 ± 0.057	4.22 ± 0.18	0.307 ± 0.006
AP	Noise	21.99 ± 0.68	0.939 ± 0.008	15.63 ± 0.33	0.830 ± 0.005
AP + **AAT**	Noise	20.64 ± 1.25	0.908 ± 0.007	16.93 ± 0.45	0.872 ± 0.027
AP	Noise or RIR	15.90 ± 0.46	0.850 ± 0.017	8.78 ± 0.22	0.572 ± 0.006
AP + **AAT**	Noise or RIR	12.25 ± 0.48	0.753 ± 0.022	6.32 ± 0.09	0.457 ± 0.008
AP	Noise and RIR	10.52 ± 0.58	0.662 ± 0.034	4.46 ± 0.21	0.316 ± 0.022
AP + **AAT**	Noise and RIR	7.95 ± 0.12	0.528 ± 0.010	3.25 ± 0.05	0.241 ± 0.007
Augment both segments					
P	Noise	19.15 ± 1.71	0.877 ± 0.017	16.65 ± 1.57	0.858 ± 0.022
P + **AAT**	Noise	17.31 ± 1.73	0.863 ± 0.011	18.81 ± 1.34	0.904 ± 0.020
P	Noise or RIR	11.31 ± 0.75	0.684 ± 0.033	9.14 ± 0.38	0.597 ± 0.029
P + **AAT**	Noise or RIR	9.17 ± 0.23	0.594 ± 0.007	7.88 ± 0.49	0.522 ± 0.018
P	Noise and RIR	5.82 ± 0.11	0.407 ± 0.003	4.62 ± 0.06	0.329 ± 0.008
P + **AAT**	Noise and RIR	5.26 ± 0.03	0.378 ± 0.009	4.11 ± 0.13	0.298 ± 0.007
AP	Noise	18.82 ± 1.13	0.895 ± 0.012	15.80 ± 1.26	0.833 ± 0.022
AP + **AAT**	Noise	18.75 ± 1.61	0.886 ± 0.022	15.61 ± 0.56	0.849 ± 0.023
AP	Noise or RIR	10.93 ± 0.28	0.687 ± 0.015	7.65 ± 0.08	0.535 ± 0.005
AP + **AAT**	Noise or RIR	9.06 ± 0.58	0.608 ± 0.013	6.52 ± 0.44	0.480 ± 0.027
AP	Noise and RIR	5.65 ± 0.42	0.401 ± 0.024	4.31 ± 0.13	0.305 ± 0.008
AP + **AAT**	Noise and RIR	4.96 ± 0.12	0.356 ± 0.007	3.47 ± 0.11	0.246 ± 0.013

TABLE III
THE EFFECT OF THE VALUE OF λ ON SPEAKER VERIFICATION PERFORMANCE, USING ADDITIVE NOISE AND ROOM REVERBERATION. RESULTS ON THE VOXCELEB1 TEST SET, USING A RESNET34 MODEL TRAINED ON VOXCELEB2

Objective	Test dataset λ	VoxCeleb1 EER (%)	VoxCeleb1 MinDCF	VOiCES EER (%)	VOiCES MinDCF
Augment both segments					
Angular Prototypical	0	9.56 ± 0.18	0.511 ± 0.011	5.65 ± 0.42	0.401 ± 0.024
Angular Prototypical + **AAT**	1	8.89 ± 0.09	0.476 ± 0.006	5.32 ± 0.19	0.361 ± 0.014
Angular Prototypical + **AAT**	3	**8.65 ± 0.14**	0.469 ± 0.008	5.05 ± 0.10	0.367 ± 0.012
Angular Prototypical + **AAT**	10	8.72 ± 0.12	**0.454 ± 0.013**	**4.96 ± 0.12**	**0.356 ± 0.007**

of 7.19%, outperforming all comparable works by a significant margin.

A similar trend is observed in VOiCES dataset results (Table II), on which the models trained with AAT outperform the counterparts without. This demonstrates that the models trained using AAT generalise well to unseen domains, as well as the dataset that the models have been trained on.

The effect of λ. Speaker recognition performance for various values of the AAT loss weight λ is reported in Table III. The augmentation process and the learning objective are fixed in these experiments. Applying the AAT improves the performance in both datasets. $\lambda = 3$ shows the best performance for VoxCeleb, and $\lambda = 10$ for VOiCES.

B. Semi-Supervised Training

Table IV reports the results of the fine-tuned models on the VoxCeleb1 test set. The semi-supervised models outperform the self-supervised counterparts by a large margin, and also show strong performance compared to the model trained using full supervision on VoxCeleb1 alone. Models trained with AAT consistently outperform the models without for various

TABLE IV
RESULTS ON THE VOXCELEB1 AND VOICES TEST SETS, USING A ECAPA-TDNN MODEL PRE-TRAINED ON VOXCELEB2 WITH SELF-SUPERVISION AND FINE-TUNED ON VARIOUS SUBSETS OF VOXCELEB1 WITH SPEAKER LABELS

Test dataset	VoxCeleb1		VOiCES	
Pre-training objective	**EER (%)**	**MinDCF**	**EER (%)**	**MinDCF**
Fine-tuned using all sessions from 1,211 speakers (148,642 utterances)				
No pretraining	2.38 ± 0.08	0.175 ± 0.003	3.24 ± 0.30	0.260 ± 0.014
Angular Prototypical	1.85 ± 0.02	0.137 ± 0.001	2.52 ± 0.05	0.202 ± 0.004
Angular Prototypical + **AAT**	**1.64 ± 0.03**	**0.135 ± 0.006**	**2.48 ± 0.04**	**0.187 ± 0.005**
Fine-tuned using all session from 500 speakers (60,559 utterances)				
No pretraining	5.19 ± 0.04	0.347 ± 0.007	5.13 ± 0.45	0.4164 ± 0.026
Angular Prototypical	2.72 ± 0.02	0.200 ± 0.004	2.68 ± 0.06	0.1868 ± 0.001
Angular Prototypical + **AAT**	**2.60 ± 0.04**	**0.187 ± 0.003**	**2.55 ± 0.10**	**0.1840 ± 0.004**
Fine-tuned using up to 4 sessions from 500 speakers (13,411 utterances)				
No pretraining	8.73 ± 0.02	0.542 ± 0.005	8.21 ± 0.33	0.6493 ± 0.027
Angular Prototypical	3.76 ± 0.02	**0.283 ± 0.001**	3.85 ± 0.03	0.3031 ± 0.007
Angular Prototypical + **AAT**	**3.62 ± 0.02**	0.286 ± 0.006	**3.38 ± 0.19**	**0.2725 ± 0.006**

TABLE V
RESULTS ON THE CN-CELEB TEST SET, USING A ECAPA-TDNN MODEL PRE-TRAINED ON VOXCELEB2 WITH SELF-SUPERVISION

Test dataset	CN-Celeb1	
Pre-training objective	**EER (%)**	**MinDCF**
No supervised fine-tuning		
Angular Prototypical + **AAT**	14.54 ± 0.08	0.658 ± 0.005
Fine-tuned on VoxCeleb1		
Angular Prototypical + **AAT**	13.46 ± 0.35	0.574 ± 0.009
Fine-tuned on CN-Celeb1		
No pretraining	11.01 ± 0.17	0.510 ± 0.004
Angular Prototypical	8.87 ± 0.18	0.423 ± 0.002
Angular Prototypical + **AAT**	**8.71 ± 0.04**	**0.422 ± 0.002**

amounts of fine-tuning data. This verifies that pre-trained models with AAT provide stronger initialisation in the fine-tuning scenario.

Table V reports the results when the pre-trained models are fine-tuned with data from different domains. Here, the model is trained on VoxCeleb2, but fine-tuned on the development set of Chinese celebrities dataset CN-Celeb1 and evaluated on the test set of CN-Celeb1. Even in this scenario, the self-supervised pre-training on the non-target domain dataset helps to improve performance.

The effect of layer freezing. The popular *wav2vec 2.0* [5] training strategy freezes the first convolutional blocks during fine-tuning. In other tasks, models fine-tuned with fixed layers perform slightly worse compared to models that are fully fine-tuned [9], [19], [21]. We conduct experiments that freeze different layers to find the best strategy in speaker verification. The self-supervised model with AAT is used to initialise the training, and the results are compared using VoxCeleb1 and CN-Celeb1.

We compare four fine-tuned models, a model without any frozen layers (i.e. fine-tuning all layers), a model with freezing the first convolutional layer of ECAPA-TDNN, a model with freezing from the first input layer to the first convolutional block, and a model with freezing from the input layer to the second convolutional block. In most cases, fine-tuning all layers without freezing performs better, as shown in Table VI.

C. Human Experiments

Metrics. For the human benchmark, we use two additional metrics – Area Under Receiver Operating Characteristic curve (AUROC), and binary classification accuracy – in addition to the Equal Error Rates (EER). EER and AUROC are obtained by interpolating the ROC curve between the points for the 5 discrete scores. Binary classification accuracy is the most intuitive and fair metric for humans, since binary decision from each pair is exactly the same task that they have been asked to perform. The score of 3 (borderline) is assigned to the positive class for both AMT and experts, since this gives a better accuracy. In reality, the annotators only used the borderline option very few times. To compute the binary classification accuracy of our self-supervised ResNet model (ResNet), we set the threshold tuned on the validation set that does not overlap with the test set in the table.

Discussion. Table VII shows the speaker verification performance of the human annotators. We also report four ROC curves based on the annotation done by the experts in the supplementary materials (Fig. 2). It can be seen that the annotations of the experts are far more accurate compared to the crowdworkers on AMT. It is also notable that the variance between the performance of the four expert annotators is relatively small.

We observe that our ResNet model outperforms the human benchmark. It is difficult for humans to match the performance of the deep learning models on the pairwise verification task. In particular, human annotators have difficulty in matching voices recorded in different environments. However, humans can sometimes use wider context in conversations or auxiliary information such as speaker's faces, which would improve the human performance in real-world scenarios.

TABLE VI
IMPACT OF FREEZING LAYERS DURING FINE-TUNING. RESULTS ON VOXCELEB1 AND CN-CELEB1 TEST SETS, FINE-TUNED USING THE RESPECTIVE DATASETS

Test dataset	VoxCeleb1		CN-Celeb1	
Frozen layer	**EER (%)**	**MinDCF**	**EER (%)**	**MinDCF**
No freezing (re-train all layers)	**1.64 ± 0.003**	**0.135 ± 0.006**	**8.71 ± 0.04**	0.422 ± 0.002
Freeze the first layer	1.64 ± 0.015	0.140 ± 0.002	8.74 ± 0.05	**0.414 ± 0.003**
+ the first convolution block	1.80 ± 0.037	0.149 ± 0.006	8.75 ± 0.04	0.423 ± 0.002
+ the second convolution block	1.84 ± 0.010	0.150 ± 0.004	8.74 ± 0.04	0.426 ± 0.002

TABLE VII
SPEAKER VERIFICATION PERFORMANCE OF DIFFERENT METHODS ON SUBSETS OF THE VOXCELEB1 TEST SET. **SS-RESNET**: OUR SELF-SUPERVISED RESNET MODEL TRAINED USING THE AP + AAT LOSS

	Metric	Verification EER (%)			AUROC (%)			Binary Classification Acc. (%)		
Test Set	**# Pairs**	**AMT**	**Experts**	**SS-ResNet**	**AMT**	**Experts**	**SS-ResNet**	**AMT**	**Experts**	**SS-ResNet**
Set A	500	25.75	16.53	8.10	79.46	89.28	97.08	73.80	82.20	91.40
Set B	500	25.63	15.70	8.75	82.33	90.96	97.73	74.40	86.00	92.00
Set C	500	22.59	17.78	9.01	81.45	88.61	97.14	75.40	82.20	90.40
Set D	500	25.91	13.98	8.76	78.34	89.78	97.30	72.60	86.40	91.40
All	2,000	26.51	15.77	8.50	79.60	89.65	97.32	74.10	84.20	91.30

V. CONCLUSION

In this paper, we proposed an augmentation adversarial training strategy to train effective speaker embeddings with self-supervision. The method exploits an augmentation classifier and gradient reversal layer to prevent the speaker embedding extractor from learning the channel information. The experiments on the VoxCeleb, CN-Celeb and VOiCES datasets demonstrate state-of-the-art performance in self-supervised and semi-supervised speaker recognition.

APPENDIX

Listing 1: PyTorch-style pseudocode of AAT

```
## Let batch is (N,3,T) dimensional tensor. N is batch_size, T is length of utterance, and D is size of embedding.
## batch[i, 0] and batch[i, 1] : different segments with same augmentation
## batch[i, 1] and batch[i, 2] : same segment with different augmentation
## netspk is speaker embedding extractor and netaug is augmentation classifier.

spk_optimizer  = optim.Adam(netspk.parameters())
aug_optimizer  = optim.Adam(netaug.parameters())

for batch in loader:
    feat = netspk.forward(batch)  # feat size : (N,3,D)

    # Discriminator Training
    aug_optimizer.zero_grad()
    out_a, out_s, out_p = feat[:,0,:].detach(), feat[:,1,:].detach(), feat[:,2,:].detach()
    conf_input = torch.cat((torch.cat((out_a,out_s),dim=1), torch.cat((out_a,out_p),dim=1)),dim=0)
    conf_output = netaug.forward(conf_input)
    conf_labels = torch.LongTensor([1] * N + [0] * N)

    aug_loss  = torch.nn.BCELoss(conf_output, conf_labels)
    aug_loss.backward()
    aug_optimizer.step()

    # Embedding training
    spk_optimizer.zero_grad()
    conf_input = torch.cat((torch.cat((feat[:,0,:],feat[:,1,:]),dim=1),torch.cat((feat[:,0,:],feat[:,2,:]),dim=1)),dim=0)
    conf_input = RevGrad(conf_input)  # reversing gradients
    conf_output = netaug.forward(conf_input)
    aat_loss = torch.nn.BCELoss(conf_output, conf_labels)
    spk_loss = SpkCriterion(feat[:, [0,2], :])  # prototypical or angular prototypical loss
    loss = spk_loss + lambda * aat_loss
    loss.backward()
    spk_optimizer.step()
```

ACKNOWLEDGMENT

We would like to thank Shinji Watanabe for helpful advice and discussions.

REFERENCES

[1] "NIST 2018 speaker recognition evaluation plan," 2018, See Section 3.1. Accessed: Jul. 31, 2020. [Online]. Available: https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf
[2] P. Anand, A. K. Singh, S. Srivastava, and B. Lall, "Few shot speaker recognition using deep neural networks," 2019, *arXiv:1904.08775*.
[3] R. Arandjelović and A. Zisserman, "Look, listen and learn," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 609–617.
[4] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 15535–15545.
[5] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2Vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 12449–12460.
[6] G. Bhattacharya, J. Alam, and P. Kenny, "Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6041–6045.
[7] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6226–6230.
[8] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, "Discriminatively trained probabilistic linear discriminant analysis for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 4832–4835.
[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[10] J. S. Chung, J. Huh, and S. Mun, "Delving into VoxCeleb: Environment invariant speaker recognition," in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, 2020, pp. 349–356.
[11] J. S. Chung et al., "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020, pp. 2977–2981.
[12] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.
[13] J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in *Proc. Asian Conf. Comput. Vis. Workshop Multi-View Lip-Reading*, 2016, pp. 251–263.
[14] S.-W. Chung, H. G. Kang, and J. S. Chung, "Seeing voices and hearing voices: Learning discriminative embeddings using cross-modal self-supervision," in *Proc. Interspeech*, 2020, pp. 3486–3490.

[15] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.
[16] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 1–5.
[17] Y. Fan et al., "CN-celeb: A challenging chinese speaker recognition dataset," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7604–7608.
[18] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.
[19] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.
[20] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," 2018, *arXiv:1807.08312*.
[21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.
[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[23] N. Inoue and K. Goto, "Semi-supervised contrastive learning with generalized contrastive loss and its application to speaker recognition," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2020, pp. 1641–1646.
[24] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Speaker Odyssey*, 2010, vol. 14.
[25] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, "The ins and outs of speaker recognition: Lessons from VoxSRC 2020," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5809–5813.
[26] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular softmax loss for end-to-end speaker verification," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, 2018, pp. 190–194.
[27] C. Luu, P. Bell, and S. Renals, "Channel adversarial training for speaker verification and diarization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7094–7098.
[28] P. Matějka et al., "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 4828–4831.
[29] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The speakers in the wild (SITW) speaker recognition database," in *Proc. Interspeech*, 2016, pp. 818–822.
[30] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6707–6717.
[31] A. Nagrani, J. S. Chung, S. Albanie, and A. Zisserman, "Disentangled speech embeddings using cross-modal self-supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6829–6833.
[32] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.
[33] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
[35] D. Povey et al., " The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2011.
[36] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proc. Interspeech*, 2019, pp. 1153–1157.
[37] C. Richey et al., "Voices obscured in complex environmental settings (VOiCES) corpus," in *Proc. Interspeech*, 2018, pp. 1566–1570.
[38] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6006–6010.
[39] A. Rouditchenko, H. Zhao, C. Gan, J. McDermott, and A. Torralba, "Self-supervised audio-visual co-segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 2357–2361.
[40] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.
[41] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," 2015, *arXiv:1510.08484*.
[42] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN embeddings for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5329–5333.
[43] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.
[44] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 4068–4076.
[45] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.
[46] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4879–4883.
[47] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. Brudno, "Centroid-based deep metric learning for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 3652–3656.
[48] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 4889–4893.
[49] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2019, pp. 1652–1656.
[50] R. Zhang, P. Isola, and Alexei A. Efros, "Colorful image colorization," in *Proc. IEEE Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 649–666.

Jingu Kang received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 2018, where he is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering. His research interests include self-supervised learning, automatic speaker verification, and automatic speech recognition.

Jaesung Huh received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2018. He is currently working toward the D.Phil. degree with the Department of Engineering Science, University of Oxford, Oxford, U.K. His research interests include audio-visual source separation, audio-visual action recognition, speaker verification, and diarization.

Hee Soo Heo received the Ph.D. degree in computer science from the University of Seoul, Seoul, South Korea, in 2019. Since 2020, he has been a Research Scientist with Naver Corporation. His main research interests include speaker recognition and audio anti-spoofing.

Joon Son Chung received the D.Phil. degree in engineering science from the University of Oxford, Oxford, U.K. He is currently an Assistant Professor with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is directing research in speech processing, computer vision, and machine learning.