The following publication L. Li, M. -W. Mak and J. -T. Chien, "Contrastive Adversarial Domain Adaptation Networks for Speaker Recognition," in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 5, pp. 2236-2245, May 2022 is available at https://doi.org/10.1109/TNNLS.2020.3044215.

Contrastive Adversarial Domain Adaptation Networks for Speaker Recognition

Longxin LI[1], Man-Wai MAK[1], and Jen-Tzung CHIEN[2]

[1]Dept. of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR
[2]Dept. of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

***Abstract*—Domain adaptation aims to reduce the mismatch between the source and target domains. A domain-adversarial network (DAN) has been recently proposed to incorporate adversarial learning into deep neural networks to create a domain-invariant space. However, DAN's major drawback is that it is difficult to find the domain-invariant space by using a single feature extractor. In this paper, we propose to split the feature extractor into two contrastive branches, with one branch delegating for the class-dependence in the latent space and another branch focusing on domain-invariance. The feature extractor achieves these contrastive goals by sharing the first and the last hidden layers but possessing decoupled branches in the middle hidden layers. For encouraging the feature extractor to produce class-discriminative embedded features, the label predictor is adversarially trained to produce equal posterior probabilities across all of the outputs instead of producing one-hot outputs. We refer to the resulting domain adaptation network as "contrastive adversarial domain adaptation network (CADAN)". We evaluated the embedded features' domain-invariance via a series of speaker identification experiments under both clean and noisy conditions. Results demonstrate that the embedded features produced by CADAN lead to a 33% improvement in speaker identification accuracy compared with the conventional DAN.**

***Keywords*— Domain adaptation; domain invariance; speaker recognition; domain adversarial networks; adversarial learning.**

I. Introduction

Speaker recognition is to verify or identify the identities of speakers by analyzing the acoustic characteristics of their speech [1]. I-vectors [2] and x-vectors [3] have proved to be very successful speaker embeddings for speaker recognition. Using probabilistic linear discriminative analysis (PLDA) [4] as the backend, the channel noise component in the i-vectors or x-vectors can be significantly suppressed.[1] However, i-vectors/PLDA systems assume that the training and test data follow the same distribution, which implies that any mismatch between the training and deployment environments may severely degrade the performance [5]–[8]. Earlier research [9] has demonstrated that when there is a severe mismatch between the training (out-of-domain) and deployment (in-domain) environments, the amount of in-domain data has a large impact on the performance of i-vector/PLDA systems.

This work was partially supported by RGC of Hong Kong SAR, Grant No. PolyU152113/17E, and Taiwan MOST Grant No. 109-2634-F-009-024.

[1]Because the methods described in this paper are equally applied to both i-vector/PLDA and x-vector/PLDA frameworks, we will use "i-vector" in the sequel for terminology simplicity.

In the literature, most of the domain adaptation methods decouple the speaker embedding process and the backend scoring process, i.e., domain adaptation is applied either during or after speaker embedding (see [1] for a comprehensive review). The advantage of decoupling is that embedding-level domain adaptation can be applied to whatever backend classifiers. Similarly, scoring-level domain adaptation can be applied to whatever speaker embeddings. This decoupling strategy makes the domain adaptation methods more versatile.

To address the domain mismatch problem, Garcia-Romero and McCree [5] proposed to estimate the within-speaker and between-speaker variabilities by treating them as random variables and used the maximum *a posteriori* (MAP) adaptation to compute these parameters on the basis of the labelled in-domain data. These covariances can also be treated as latent variables [7] whose joint posterior distribution can be factorized by using the variational Bayes algorithm. Thus, the point estimates for scoring the in-domain data are computed from the factorized distribution. These earlier methods correspond to *supervised* domain adaptation because they require the in-domain training data to have speaker labels.

One approach to dealing with unlabelled data is to hypothesize speaker labels by unsupervised clustering of in-domain data [10], [11]. With the hypothesized speaker labels, an in-domain PLDA model can be trained. An adapted PLDA model can be obtained by using the interpolated covariance matrices of the out-of-domain PLDA model and the in-domain PLDA model [10]. The drawback of the clustering approach is that the number of speakers in the in-domain data is usually unknown.

Another way to perform unsupervised adaptation is to find a domain-invariant space from several datasets, each collected from one domain. For example, Aronowitz [12], [13] proposed an inter-dataset variability compensation (IDVC) algorithm to reduce the mismatch between datasets. The algorithm was further extended in [14]. IDVC assumes that within the i-vector space there is a low-dimensional subspace that is more sensitive to dataset mismatch. Therefore, the goal of IDVC is to find this subspace and remove it from all of the i-vectors. To find this subspace, IDVC either divides a big heterogeneous dataset into a number of source-dependent subsets or makes use of multiple datasets with each dataset represents one source. Another approach is to normalize the covariances of out-of-domain i-vectors [15], which has similar notion as within-class covariance normalization [8] but without using speaker labels. The authors in [15] named the method as

dataset invariant covariance normalization (DICN). Recently, the authors in [16], [17] used maximum mean discrepancy (MMD) among multiple datasets as a loss function for training an autoencoder so that domain-invariant i-vectors can be extracted from its middle layer. Unlike IDVC and DICN, the MMD loss reduces domain mismatches beyond the second order statistics.

Domain adversarial training (DAT) [18]–[20] is a state-of-the-art domain adaptation method for domain adaptation. The method adversarially trains a set of networks comprising a feature extractor, a label predictor and a domain discriminator. The three components work cooperatively but also challenge each other to form a domain-invariant space with maximum class information. In [21], Wang *et al*. demonstrated the effectiveness of domain adversarial training for speaker recognition through creating a domain-invariant and speaker-discriminative space. PLDA was used as the back-end to score the vectors extracted from the adversarial network. The results suggest that DAT outperforms other unsupervised domain adaptation methods including IDVC [12], [13], DICN [15], and matrix interpolation [10].

In this paper, we propose a contrastive adversarial domain adaptation network (CADAN) that utilizes adversarial learning to create a domain-invariant space with maximum speaker information. Features extracted from this space can replace the conventional i-vectors for speaker recognition. Unlike the conventional domain adversarial network (DAN), we separate the feature extractor in DAN into two parts, one part for maximizing the class information in the domain-invariant space and the other part minimizes the domain information. The weights of the two parts are separately updated to achieve these two contrastive goals. Also, unlike the conventional DAN in which the label predictor is trained to minimize the cross-entropy loss, we purposely weaken the capability of the label predictor in classifying speakers. This has the effect of forcing the feature extractor to work harder to produce more class discriminative features. Because our class-label predictor aims to make the life of the feature extractor harder as opposed to making it easier, we name it as the *fuzzifier*.

In addition to comparing with DAN, this paper also uses t-distributed stochastic neighbor embedding (t-SNE) [22] plots to illustrate the domain-invariance and class discrimination of the embedded features created by the CADAN during adversarial training. Experimental results on NIST 2012 SRE demonstrate that the CADAN can achieve nearly ideal domain adaptation for gender mismatch on speaker identification and outperforms state-of-the-art domain adversarial networks in both clean and noisy environments.

II. BACKGROUND

A. I-vector and PLDA Framework

I-vectors [2] are compact representation of speaker utterances. The method uses factor analysis to compress the frame-based acoustic information of an utterance into a low-dimensional vector called the i-vector. Mathematically, given the acoustic feature vectors of an utterance, its MAP-adapted GMM-supervector [23] $\boldsymbol{\mu}_s$ is assumed to be generated by a factor analysis model:

$$\boldsymbol{\mu}_s = \boldsymbol{\mu} + \mathbf{T}\mathbf{w}_s, \tag{1}$$

where $\boldsymbol{\mu}$ is the universal mean vectors obtained by stacking the mean of a universal background model (UBM), $\mathbf{T}$ is a low-rank total variability matrix modeling the variabilities of speakers and channels, and $\mathbf{w}_s$ is a latent vector whose posterior mean is the i-vector. Details of the i-vector extraction process can be found in [1], [24].

Because the i-vectors contain both speaker and non-speaker (typically channel) information, it is important to suppress the non-speaker information during scoring. A state-of-the-art approach to achieving this goal is to apply supervised factor analysis on a set of training i-vectors with speaker labels to find a speaker subspace within the i-vector space. The covariances of i-vectors are considered as the sum of speaker covariances and non-speaker covariances. The method is called probabilistic linear discriminant analysis (PLDA) in the literature [4], [25].

Given a dataset comprising length normalized [26] i-vectors $\mathcal{X} = \{\mathbf{x}_{ij} \in \Re^D; i = 1, \ldots, N; j = 1, \ldots, H_i\}$ where N is the number of speakers and H_i is the number of sessions of speaker i, the PLDA model can be expressed as follows:

$$\mathbf{x}_{ij} = \mathbf{m} + \mathbf{V}\mathbf{z}_i + \boldsymbol{\epsilon}_{ij}, \tag{2}$$

where $\mathbf{z}_i \in \Re^M, i = 1, \ldots, N$, are the latent variables, $\boldsymbol{\epsilon}_{ij}$ is the residue that follows a Gaussian distribution, $\mathbf{m}$ is the global mean of i-vectors and $\mathbf{V}$ defines the speaker subspace. When the prior of $\mathbf{z}_i$'s follows a standard Gaussian and $\boldsymbol{\epsilon}_i$'s follow a Gaussian with zero mean and full covariance matrix $\boldsymbol{\Sigma}$, Eq. 2 is known as the simplified Gaussian PLDA. Its parameters can be obtained by the expectation-maximization algorithm [4], [24].

B. Adversarial Learning

Adversarial learning is a machine learning technique typically used for developing and evaluating security systems under adversarial environments and malicious attacks [27]. Recently, the technique is popularized by the success of generative adversarial networks (GANs) [28] and their enhancements that overcome the training difficulties [29], [30] and mode collapse [31]–[33].

Adversarial learning can be applied to train a DNN to create a domain-invariant space [18] or to align the class distributions of the source task and the target task [34]. Unlike GANs, these networks do not have random inputs; instead, they receive domain-dependent feature vectors as inputs. Their goal is to create a representation with minimum domain dependence. The domain adversarial network in [18] incorporates adversarial learning into deep neural networks by creating a latent space in which the domain discrepancy is suppressed while the class-dependent information is maintained. A DAN comprises three components: a feature extractor, a domain discriminator, and a label predictor. During training, the feature extractor and the label predictor are jointly trained to minimize the cross-entropy errors in the label predictor's output. Also, the feature

extractor is jointly trained with the domain discriminator. But unlike the feature-extractor–label-predictor combination, for the feature-extractor–domain-discriminator combination, the feature extractor is adversarially trained so that the resulting features maximize the loss of the domain discriminator. The adversarial learning algorithm acts like a two-player game in which the feature extractor is trained to confuse the domain discriminator that is tuned to distinguish the target domain from the source domain. The designated output at the intermediate layer of the domain-adversarial neural network is not only domain-invariant but also class discriminative.

C. Domain Mismatch in Speaker Recognition

Domain mismatch can severely degrade speaker recognition performance [5]–[7], [13], [35]. There are various causes of domain mismatch, the most prominent being the discrepancy between training and test environments arising from different channels, languages, dialects, genders, noise types, noise levels, and reverberation effects.

Gender difference is one of the most severe and obvious mismatch due to the physiological differences between male and female. A recent study [36] demonstrated that speaker verification performance can be improved by predicting the gender of an unknown speaker followed by gender-dependent scoring. In another study [11], a DNN was used for computing the posterior probabilities of genders, which were then used as mixture posteriors in a PLDA mixture model. It was shown that although the gender information could not be perfectly predicted, it is helpful for the PLDA mixture model to score the i-vectors, resulting in performance superior to a gender-independent PLDA model. The systems in these studies, however, require a gender classifier because both of their speaker embeddings and backend classifiers are gender-dependent. We advocate that gender-independent speaker recognition systems are more practical because they neither need gender information in the training data nor gender classifiers during recognition. To this end, we treat the gender-mismatch as a domain adaptation problem. By minimizing the gender effect on the speaker embeddings, it is possible to build gender-independent systems without compromising performance.

Another cause of mismatch is background noise, which could be of different types and levels. As demonstrated in [37], different levels of noise could cause the i-vectors to fall on distinct regions of the i-vector space, which motivates the use of mixture PLDA models as the backend classifier. In this paper, we focus on the mismatches caused by different levels of car noise, factory noise, and babble noise. We advocate that domain adaptation is essential for reducing the noise mismatch because, in most cases, we could not predict the type of noise that a deployed system will encounter.

III. Contrastive Adversarial Domain Adaptation

The main challenge in domain adaptation is that we need to minimize the domain information in feature vectors without affecting their class information. We propose a new contrastive adversarial domain adaptation network (CADAN) to meet this challenge. This section explains the design philosophy, architecture, and training algorithm of the CADAN.

A. Design Philosophy and Network Architecture

In the original DAN, the feature extractor is particularly hard to train because it needs to produce features that meet two contrastive objectives: maximum class discrimination and minimum domain dependency. In practice, its weights are tuned to meet the first objective but will be re-adjusted to meet the second one in the same epoch. This is in analogy to asking a person to learn two different but related tasks at the same time, which of course will not be as effective as learning one task at a time. While we may change the training strategy so that the two tasks can be learned consecutively, it is also undesirable because the network may forget the first task after learning the second one. A better approach is to delegate some task-specific neurons for the respective tasks. To this end, we propose splitting the middle hidden layers of the feature extractor network into two branches so that they become partially decoupled from each other during adversarial training. In spite of the decoupling, the two sub-networks need to cooperate with each other because for each input vector, the feature extractor needs to produce one embedded feature vector as output. Therefore, the two branches share the input layer and the output layer. The architecture is shown in Fig. 1.

In addition to the contrastive feature extractor, another key difference between the proposed architecture in Fig. 1 and the DAN is the label predictor. In DAN, the feature extractor and label predictor are jointly trained to minimize the cross-entropy of the target classes. However, in the proposed architecture, the class encoder is trained to minimize the cross-entropy but the label predictor is trained to produce equal outputs (posterior probabilities). Therefore, instead of making the predictor more capable of classifying the latent feature vectors, we make it less capable of doing so. From the label predictor perspective, the latent features become *fuzzier* after every epoch. The deliberately weakening of the label predictor will encourage the class encoder in Fig. 1 to try harder to produce more speaker discriminative features so that they can be classified correctly by the adversarially trained label predictor. Because the label predictor is adversarially trained, the embedded features become more confusable to the label classifier. Therefore, we refer to the label classifier as "Fuzzifier".

In the proposed approach, the feature extractor G is split into a domain suppressor G_{dom} and a class encoder G_{cls}. As shown in Fig. 1, the neurons in the feature extractor are separated into the blue group G_{cls}, which is to be trained with the fuzzifier F to maximize class discrimination, and the green group, which is to be trained with the domain discriminator. Because of the different objectives when training the weights (blue) for encoding class-discriminative information and the weights (green) for domain discrimination, both G_{cls} an G_{dom} become better in performing their respective tasks. Without the separate structure, training will become unstable if the weights are updated twice for different purposes in each epoch.

B. Training Algorithm

The training of F and G_{cls} in Fig. 1 are as follows:

$$\text{Train } F : \min_{F} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} \frac{1}{K} \log F(G(\mathbf{x}))_k \right] \right\} \tag{3a}$$

$$\text{Train } G_{\text{cls}} : \min_{G_{\text{cls}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \log F(G(\mathbf{x}))_k \right] \right\}, \tag{3b}$$

where $G(\mathbf{x})$ is the output of the contrastive feature extractor, $F(\cdot)_k$ is the k-th output of the fuzzifier whose designated output is the posterior of class k, and $y_{\text{cls}}^{(k)}$ is equal to 1 if $\mathbf{x}$ comes from the k-th class; otherwise it is equal to 0. Unlike ordinary DAN in which the targets of the class classifier are in one-hot format, in CADAN, the targets of F in Eq. 3(a) are set to $[\frac{1}{K}, \ldots, \frac{1}{K}]^{\mathsf{T}}$. It can be shown that the minimum of the cross-entropy in Eq. 3(a) occurs when $F(G(\mathbf{x}))_k = \frac{1}{K}$ for all k. When this happens, the encoded vectors $\hat{\mathbf{z}} = G(\mathbf{x})$ will be most confusable to the fuzzifier. During training, the classification ability of F will keep on weakening. The weak F will make the class encoder G_{cls} to work harder to produce class-discriminative features to reduce the cross-entropy in Eq. 3(b).

The encoder G_{dom} in Fig. 1 aims to make the embedded vectors $\hat{\mathbf{z}}$'s domain invariant. This can be achieved by the following optimization:

$$\text{Train } D : \min_{D} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{m=1}^{M} y_{\text{dom}}^{(m)} \log D(G(\mathbf{x}))_m \right] \right\} \tag{4a}$$

$$\text{Train } G_{\text{dom}} : \min_{G_{\text{dom}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{m=1}^{M} \frac{1}{M} \log D(G(\mathbf{x}))_m \right] \right\}, \tag{4b}$$

where $y_{\text{dom}}^{(m)} = 1$ when $\mathbf{x}$ comes from domain m; otherwise $y_{\text{dom}}^{(m)} = 0$. The weights of G_{dom} are updated to obtain a domain invariant space so that the encoded vectors $\hat{\mathbf{z}}$'s become confusable to the discriminator D. To achieve this, the targets of D in Eq. 4b are set to $[1/M, \ldots, 1/M]^{\mathsf{T}}$. The domain discriminator D is trained to best differentiate these confusable vectors into different domains. The pseudo-code for training a CADAN is shown in Algorithm 1, where for each mini-batch, D, G_{dom}, G_{cls}, and F are trained consecutively. The training algorithm optimizes the domain discriminator, domain suppressor, class encoder and fuzzifier in each learning epoch. The class encoder is updated with R steps within an epoch.

IV. Experimental Setup

To evaluate the effectiveness of the CADAN in suppressing domain mismatch, we employed it in a speaker identification task in which genders are considered as domains and speaker identities are considered as classes (see Section II-C for the rationale of reducing gender mismatch via domain adaptation). Therefore, K and M in Fig. 1 correspond to the number of speakers and the number of domains (or datasets), respectively.

A. Speech Data and Acoustic Features

Speech files from NIST 2004–2012 Speaker Recognition Evaluation (SRE04–12) were used as the training and test datasets. Car noise and factor noise from NOISEX-92 [38] were added to the speech files of SRE04–12 at an SNR of 6dB. Also, babble noise from the PRISM dataset [39] was added to the speech files of SRE04–12 at SNR of 0dB, 6dB, and 15dB. Each dataset was first divided into male and female subsets. The speech files of each speaker were further split into training and test sets to ensure that the speakers in the test utterances must exist in the training set.

Because SRE04–12 contains telephone conversations and interviews, this way of splitting the data can also ensure that the contexts of the training utterances are totally different from those of the test utterances. A 2-channel voice activity detector (VAD) [40] was applied to remove silence regions. We follow the standard signal processing pipeline for extracting acoustic features from utterances [1]. Specifically, for each speech frame, 19 MFCCs together with energy plus their first and second derivatives were computed, followed by cepstral mean normalization [41] and feature warping [42] with a window size of three seconds. A 60-dim acoustic vector was extracted every 10ms, using a Hamming window of 25ms.

B. I-Vector Extraction

A subset of the telephone and microphone speech files in SRE05–10 were used for training a gender-independent UBM with 1024 mixtures. Then, MAP adaptation [43] was applied to adapt the gender-independent UBM to gender-dependent UBMs using the speech files of the respective gender as adaptation data. For each gender, a 500-factor total variability (TV) matrix ($\mathbf{T}$ in Eq. 1) was estimated. The gender-dependent TV matrices and UBMs were used for extracting gender-dependent i-vectors. Using MAP adaptation to create gender-dependent UBMs can ensure that there is a one-to-one correspondence between their Gaussians, which in turn ensures that the GMM-supervectors of both genders ($\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}$ in Eq. 1) live in the same Euclidean space. As a result, the gender-dependent i-vectors also live in the same 500-dimensional i-vector space.

C. Configuration and Training of DAN and CADAN

To ensure fair comparisons between DAN and CADAN, we kept their structure almost the same. Specifically, both of them have 500 input nodes, 3 hidden layers with 1,200 ReLU nodes in each layer, and 500 output nodes in the feature extractor. However, for CADAN, the 2nd hidden layer was split into two parts: 800 nodes for the class (speaker) encoder and 400 nodes for the domain (gender) suppressor. The ratio of 2:1 is motivated by the intuition that speaker information is more diverse than gender information, thereby requiring more nodes to encode. For both DAN and CADAN, the fuzzifier and the domain discriminator comprise two hidden layers, each with 500 ReLU nodes. The fuzzifier has 67 output nodes corresponding to 67 speakers and the domain discriminator has two output nodes.

We used the i-vectors of both genders in SRE04–10 to train a DAN and a CADAN. After training, we used the

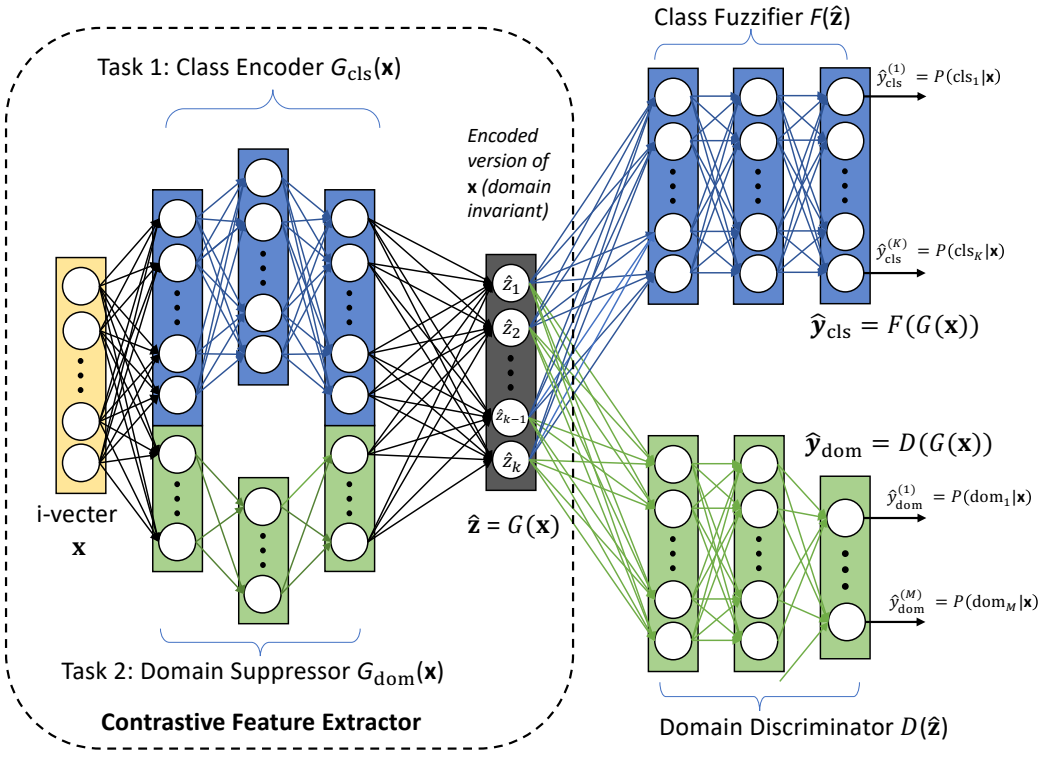


Fig. 1: Contrastive Adversarial Domain Adaptation Networks (CADAN). The blue layers constitute the adversarial networks for enhancing class information and the green layers are responsible for reducing domain mismatch. The subscript "cls" and "dom" stand for class and domain, respectively.

feature extractor network of the DAN and the contrastive feature extractor network of the CADAN to produce DAN- and CADAN-transformed vectors $\hat{\mathbf{z}}$'s for both training and test i-vectors in the datasets. The training subset of the transformed vectors were then used for training gender-dependent PLDA models. The DAN and CADAN trained by SRE04–10 correspond to the columns "SRE04–10" in Table I and Table II. To investigate the behavior of DAN and CACAN under noisy environments, babble noise was added to the telephone utterances of SRE04–12. We used the i-vectors extracted from the noise-contaminated SRE04–10 utterances to train a DAN and a CADAN, and their performance on noise-contaminated SRE12 utterances is shown in the columns "Noisy SRE12" in Table I and Table II.

D. PLDA Training and Scoring

A pre-processing step was applied to the transformed vectors before they were used for training PLDA models. Specifically, the DAN- and CADAN-transformed vectors were subjected to within-class covariance normalization [44], length normalization [26], and linear discriminant analysis (LDA). The LDA reduces the dimension of the transformed vectors to 200. The WCCN and LDA matrices are gender-dependent and were estimated from the transformed i-vectors in SRE05–10. Similarly, the WCCN and LDA matrices for "Noisy SRE12" were obtained from the i-vectors of noise contaminated speech in SRE12. The pre-processed vectors were then used for training condition-dependent (clean or noisy) and gender-dependent PLDA models.

In the testing phase, test i-vectors were transformed by the feature extractors of DAN and CADAN, respectively, followed by WCCN, length normalization, and LDA. The test i-vector pairs were then passed to the corresponding PLDA model for scoring. Fig. 2 shows the DAN/CADAN transformation, vector pre-processings and PLDA scoring.

Because each speaker has multiple training sessions (i-vectors), the speaker ID of each test i-vector was identified based on the maximum average PLDA scores (averaged across all training sessions of each speaker) with respect to all speakers in the dataset.

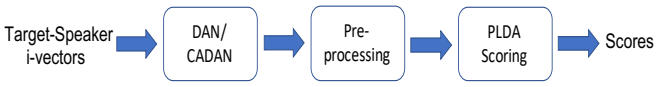


Fig. 2: Transformation of i-vectors by the feature extractor of DAN or CADAN and pre-processing for PLDA scoring. The transformation is considered as domain adaptation in this work.

V. Results and Discussions

A. Comparing DAN and CADAN

While the DAN and CADAN were trained on the speech (i-vectors) of both genders, the UBMs, T-matrices, and PLDA models are gender-dependent. With these gender-dependent PLDA models, we could have three kinds of experiments: (1) same-gender, (2) cross-gender, and (3) mix-gender.

		Out-of-Domain				In-Domain			
		SRE04–10		Noisy SRE12		SRE04–10		Noisy SRE12	
Gender of PLDA models		male	female	male	female	male	female	male	female
Gender of test i-vectors		female	male	female	male	male	female	male	female
I-Vector Transformation Method	None	0.6115	0.6281	0.5439	0.3248	0.8758	0.9375	**0.8619**	**0.6890**
	DAN	0.6337	0.6004	0.6364	0.3912	0.8417	0.9304	0.6932	0.5681
	CADAN	**0.6987**	**0.6723**	**0.7343**	**0.6556**	**0.8887**	**0.9468**	0.8307	0.6541

TABLE I: Speaker identification accuracies on SRE04–10 and noisy SRE12 with and without i-vector transformation under gender-match and gender-mismatch scenarios. Out-of-domain (in-domain) means that the gender of PLDA models is the same as (different from) that of the test i-vectors. The column labels "SRE04–10" and "Noisy SRE12" denote the source of test data for obtaining the identification accuracies. See Section IV-C for the noise contamination procedure.

		SRE04–10		Noisy SRE12	
Gender of PLDA models		male	female	male	female
Gender of test i-vectors		Both		Both	
I-Vector Transformation Method	None	0.6687	0.5770	0.6494	0.5391
	DAN	0.6512	0.6051	0.6264	0.5512
	CADAN	**0.7134**	**0.6823**	**0.6807**	**0.5691**

TABLE II: Speaker identification accuracies on SRE04–10 and noisy SRE12 with and without i-vector transformation when the test i-vectors come from both genders but the PLDA model belongs to one gender only.

		SRE04–10 (Male, Microphone Speech)					
Training Domain of PLDA models		Factory	Factory	Babble	Babble	Car	Car
Test Domain of i-vectors		Car	Babble	Factory	Car	Babble	Factory
I-Vector Transformation Method	None	0.7446	0.7523	0.7677	0.7119	**07371**	0.7393
	DAN	0.7213	0.7551	0.6921	0.7022	0.7130	**0.7447**
	CADAN	**0.7594**	**0.7658**	**0.7807**	**0.7232**	0.7113	0.6986

TABLE III: Speaker identification accuracies on the male SRE04–10 microphone speech contaminated by car, factory and babble noise at an SNR of 6dB.

		SRE04–10 (Male, Microphone Speech)					
Training Domain of PLDA models		15 dB	15 dB	6 dB	6 dB	0 dB	0 dB
Test Domain of i-vectors		0 dB	6 dB	15 dB	0 dB	6 dB	15 dB
I-Vector Transformation Method	None	0.6607	**0.8012**	0.7397	0.6438	**0.6412**	0.6303
	DAN	**0.6312**	0.8177	0.7212	0.6551	0.6405	0.6191
	CADAN	0.6487	0.7871	**0.7447**	**0.6617**	0.6377	**0.6544**

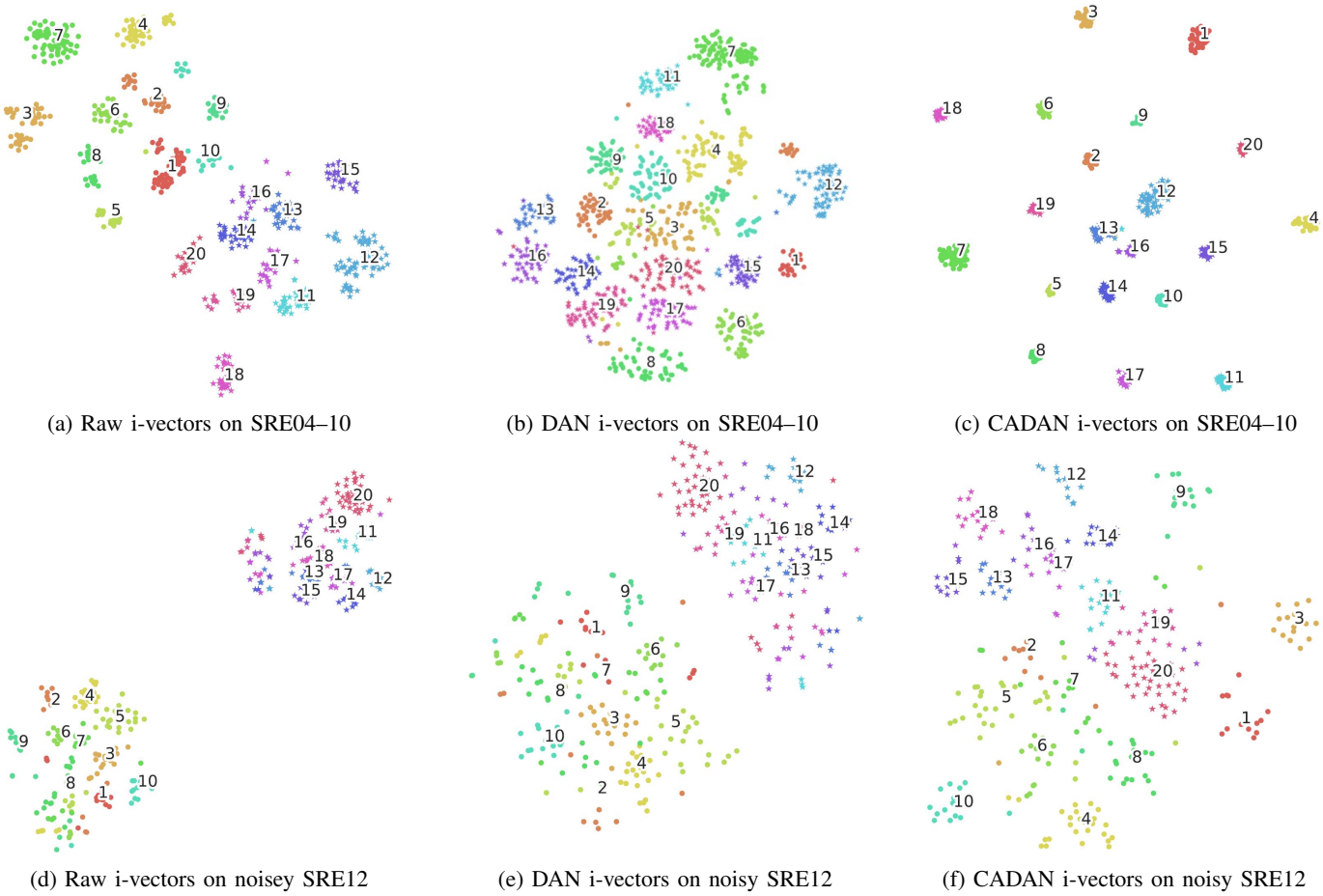
TABLE IV: Speaker identification accuracies on the male SRE04–10 microphone speech contaminated by babble noise at SNR of 0dB, 6dB, and 15dB.

1) *Same-gender Experiments*. The PLDA models were trained and scored on the DAN- and CADAN-transformed i-vectors derived from the same gender.
2) *Cross-gender Experiments*. The male PLDA models were tested on female vectors and vice versa for the female PLDA models.[2]
3) *Mix-gender Experiments*. The gender-dependent PLDA models were tested on the vectors from both genders.

Table I shows the performance of the baseline (the row with label 'None') and the DAN- and CACAN-transformed i-vectors. The baseline performance is based on an i-vector PLDA system in which the PLDA model was trained by the pre-processed i-vectors without domain adaptation.

Table I demonstrates that CADAN performs the best under the cross-gender scenario (out-of-domain columns) and performs well under the same-gender scenario (in-domain columns), although it is out-performed by the baseline under gender-match noisy conditions. The positive results reveal that contrastive-adversarial domain adaptation is capable of producing more effective features with rich speaker information. Under noisy scenarios, the CADAN demonstrates superior performance in out-of-domain data by boosting the accuracy by 33%. It is noteworthy that while adversarial

[2]It is possible to do this because the PLDA model is only a scorer; it accepts two vectors as input and computes the score of these two vectors as output. Therefore, a male PLDA model can be used for scoring female i-vectors.



(a) Raw i-vectors on SRE04–10 (b) DAN i-vectors on SRE04–10 (c) CADAN i-vectors on SRE04–10

(d) Raw i-vectors on noisey SRE12 (e) DAN i-vectors on noisy SRE12 (f) CADAN i-vectors on noisy SRE12

Fig. 3: t-SNE plots of raw i-vectors and DAN- and CADAN-transformed i-vectors derived from clean SRE04–10 utterances and noisy SRE12 utterances. I-vectors were derived from the utterances of 10 male speakers (•) and 10 female speakers (⋆). The numbers on top of each cluster are the speaker numbers (Speakers 1–10 are male and Speakers 11–20 are female) and each speaker is represented by one colour. Note that the DAN- and CADAN-itransformed i-vectors are of 500 dimensions which is the same as the dimension of the raw i-vectors.

transformation of i-vectors can increase speaker identification accuracy under gender-mismatch (out-of-domain) condition, it is counter-intuitive to apply the transformation under gender-match (in-domain) condition. This is evident by the "In-Domain" columns in Table I where not applying any transformation achieves the best performance. This result suggests that some speaker information is inevitably lost in the i-vector transformation process. Nevertheless, the loss is much smaller in CADAN than in DAN.

We further extended our experiments to gender-mixed scenarios, in which each PLDA model was trained by one gender only but tested on both genders. As shown in Table II, CADAN performs the best under all conditions.

In addition to gender mismatch, domain mismatches caused by different noise types and different noise levels were also investigated in Table III and Table IV, respectively. Table III shows that the CADAN achieved better and stabler performance in the majority of scenarios where the PLDA was trained by the noised-contaminated utterances of one kind of noise but tested on the other types of noise. Similarly, Table IV shows that the performance of CADAN is more stable under noise-mismatch scenarios.

B. *Visualization of CADAN*

To investigate the hidden causes of the better performance achieved by CADAN, we used the t-SNE software to display the i-vectors in Fig. 3. The t-SNE plots of clean SRE04–10 reveal three interesting observations. (1) There is a significant gender mismatch between the i-vectors of male and female speakers, as evident by the clear gaps in the middle of Fig. 3a and Fig. 3d that separate the two genders (• and ⋆). While Fig. 3a shows that the raw i-vectors do contain speaker information (as evident by the speaker clusters), some speakers such as Speakers 13, 14, and 16 are fairly confusable. (2) DAN is able to create a gender-invariant space, as evident by the absence of a clear gap between the two genders in Fig. 3b. However, as compared to the raw i-vectors in Fig. 3a, the feature extractor of DAN removes some of the speaker information when it attempts to make the transformed i-vectors gender indistinguishable, as evident by the larger speaker clusters in Fig. 3b. This means that DAN is not able to maximize speaker information and minimize

domain information *simultaneously*. (3) Compared with the raw and DAN-transformed i-vectors, CADAN can produce i-vectors that possess the strongest discriminative information and simultaneously suppress domain information significantly, which result in highly compact speaker clusters in Fig. 3c.

Fig. 3d shows that noise has detrimental effect on i-vectors. It not only makes the gender gap bigger, but also increases the overlapping among speaker clusters. Under noisy environments, the domain (gender) mismatch is so severe that DAN can only reduce the gender gap but fails to create a domain invariant space, as shown in Fig. 3e. On the other hand, as shown in Fig. 3f, CADAN is not only able to create a domain-invariant space but also able to reduce the cluster overlapping. This ability makes CADAN significantly outperforms raw i-vectors and DAN-transformed i-vectors in Table I under the cross-gender scenario.

Fig. 4 shows the cross-entropy loss of DAN and CADAN during the course of training. The results clearly show that CADAN enjoys faster convergence, smoother training, and lower training error as compared to DAN.

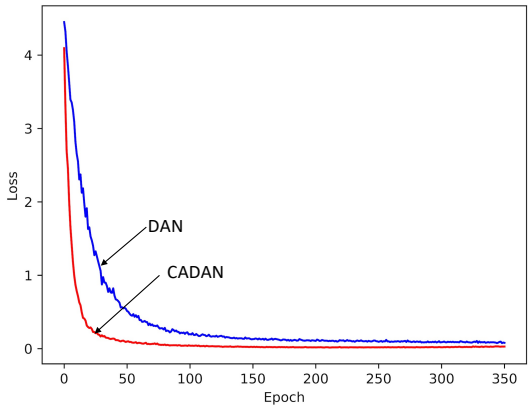
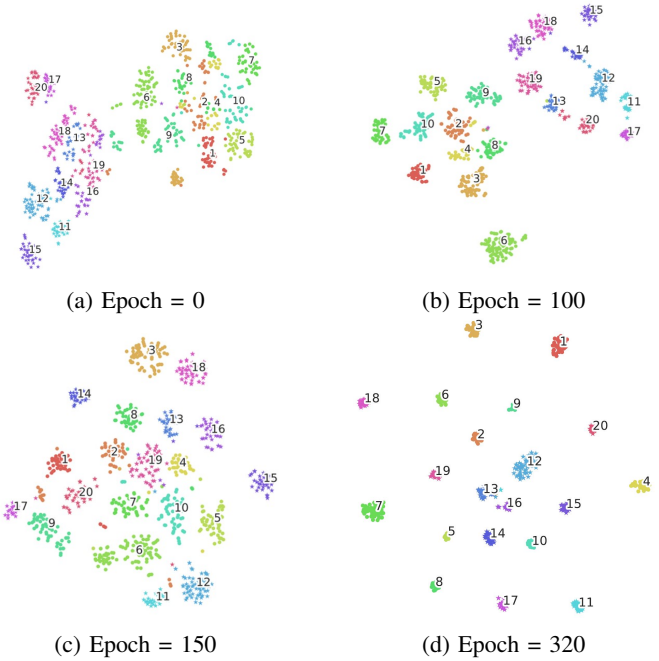


Fig. 4: The cross-entropy loss of (1) the feature extractor cum the class predictor in DAN [18] and (2) the class encoder $G_{\text{cls}}(\mathbf{x})$ cum the class fuzzifier F in CADAN. Identical learning rate (0.001) was applied in both cases.

C. Insights from the Training Process

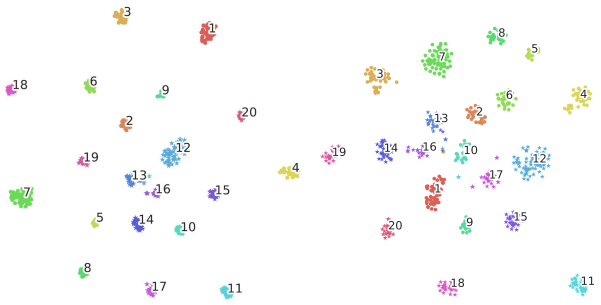
A deeper investigation was conducted to gain more insights into the training process of CADAN by plotting the intermediate transformed i-vectors at different training epochs in Fig. 5. At Epoch 0, the weights of CADAN was initialized by the Xavier initializer, which leads to scattered i-vectors in Fig. 5a. When training progresses (Fig. 5b), the fuzzifier F and class encoder G_{cls} dominate the process by minimizing intra-speaker variability but the domain mismatch remains intact. After producing a discriminative subspace, the domain discriminator D and the domain suppressor G_{dom} work on pulling the male and female groups together. At this stage (Epoch = 150), Fig. 5c, the adapted subspace with discriminative information is produced. One advantage of CADAN is that the refinement of clusters will be further conducted if training continued. At the final stage (Epoch = 320, Fig. 5d), the clusters are nearly ideal and the subspace is domain-invariant. The behavior of CADAN during the course of training reveals that it is able to response to different training objectives. In particular, with in a short training window, CADAN will either learn to perform domain adaptation or speaker discrimination, which exactly matches our original intention to design two separate feature extractors that response to different training objectives independently.



(a) Epoch = 0 (b) Epoch = 100

(c) Epoch = 150 (d) Epoch = 320

Fig. 5: t-SNE plots at different training stages of CADAN. I-vectors were derived from the utterances of 10 male speakers (•) and 10 female speakers ($\star$). The numbers on top of each cluster are the speaker numbers (Speakers 1–10 are male and Speakers 11–20 are female) and each speaker is represented by one colour.



(a) (b)

Fig. 6: t-SNE plots of transformed vectors ($\hat{\mathbf{z}}$) obtained by (a) CADAN with a fuzzifier in Fig. 1 and (b) a CADAN with the fuzzifier replaced by a speaker classifier. Refer to the caption of Fig. 5 for the meaning of markers and colors.

D. Speaker Fuzzifier Versus Speaker Classifier

Recall that the motivation of using a fuzzifier instead of a speaker classifier in CACAN is that the former is better at forcing the class encoder G_{cls} in Fig. 1 to produce more speaker discriminative latent vectors than the latter. To demonstrate that it is indeed the case, we conducted another experiment in which the fuzzifier in Fig. 1 was replaced by a speaker classifier C. The network is similar to a DAN except for the splitting of the feature extractor into two branches. The objective functions in Eq. 3a and Eq. 3b are modified as follows:

$$\text{Train } C : \min_{C} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \ \log C(G(\mathbf{x}))_k \right] \right\} \tag{5a}$$

$$\text{Train } G_{\text{cls}} : \min_{G_{\text{cls}}} \left\{ -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} \left[\sum_{k=1}^{K} y_{\text{cls}}^{(k)} \ \log C(G(\mathbf{x}))_k \right] \right\}. \tag{5b}$$

The training of G_{dom} and D in Eq. 4a and Eq. 4b remains unchanged.

Fig. 6 compares the transformed vectors obtained by the proposed CACAN and a CACAN whose fuzzifier is replaced by a speaker classifier. The result clearly shows that the fuzzifier can make the transformed vectors more speaker discriminative. A possible explanation is that minimizing the cross-entropy in Eq. 5a will make the lower layers of the speaker classifier to contain speaker information. This information will be wasted because we will only use the feature extractor to produce the transformed vectors after training. On the other hand, minimizing the cross-entropy in Eq. 3a encourages confusable input but Eq. 3b encourages speaker discriminative transformed vectors. As a result, the fuzzifier ensures that speaker information will be kept in the latent representation $\hat{\mathbf{z}}$.

In summary, the fuzzifier can force the class encoder to take all of the responsibility for producing discriminative features so that all of the usable discriminative information is encapsulated inthe class encoder by which a more discriminative subspace can be created.

VI. Conclusions

In this work, we proposed a contrastive adversarial domain adaptation network (CADAN) which achieves a significant improvement in domain adaptation for speaker recognition when compared with state-of-the-art domain adversarial network (DAN). We contributes to two major modification of the original recipe: (1) splitting the encoder into two separate networks (class encoder and domain suppressor) for different purposes and (2) replacing the classifier with a fuzzifier for enhancing the discriminative information in the encoded features $\hat{\mathbf{z}}$. A profound improvement was observed by using PLDA models to score the encoded i-vectors for speaker recognition. The visualization of the encoding process of CADAN also shows that the modified networks are more effective in producing discriminative features and suppressing domain information. While this paper has shown that CADAN can improve gender- and noise-dependent systems, it is interesting to investigate its performance on speaker recognition systems trained with gender-mix speech under a wide variety of noisy conditions in future work.

References

[1] M.-W. Mak and J.-T. Chien, *Machine Learning for Speaker Recognition*. Cambridge University Press, 2020.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

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

[4] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.

[5] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4047–4051.

[6] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 409–431.

[7] J. Villalba and E. Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2012, pp. 641–647.

[8] O. Glembek, J. Ma, P. Matejka, B. Zhang, O. Plchot, L. Burget, and S. Matsoukas, "Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4032–4036.

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[10] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 265–272.

[11] L. X. Li and M. W. Mak, "Unsupervised domain adaptation for gender-aware PLDA mixture models," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5269–5273.

[12] H. Aronowitz, "Compensating inter-dataset variability in PLDA hyper-parameters for robust speaker recognition," in *Speaker Odyssey: Speaker and Language Recognition Workshop*, 2014, pp. 282–286.

[13] ——, "Inter dataset variability compensation for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4002–4006.

[14] ——, "Inter dataset variability modeling for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5400–5404.

[15] M. H. Rahman, A. Kanagasundaram, D. Dean, and S. Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proc. Interspeech*, 2015, pp. 1017–1021.

[16] W. W. Lin, M. W. Mak, L. X. Li, and J.-T. Chien, "Reducing domain mismatch by maximum mean discrepancy based autoencoders," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 162–167.

[17] W. W. Lin, M. W. Mak, and J.-T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[19] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *arXiv preprint arXiv:1409.7495*, 2014.

Algorithm 1 Training of CADAN (Fig. 1). In Line 9, $\mathbf{W}_1$ ($\mathbf{W}_2$) contains the weights connecting G_{cls} (G_{dom}) and the output nodes of the feature extractor. In Lines 11 and 12, M is the number of domains. In Line 13, R is the number of inner iterations for training G_{cls} within an epoch.

1: **procedure** CADAN_TRAIN($\mathcal{X}, \mathcal{Y}_{\text{cls}}, \mathcal{Y}_{\text{dom}}$)
2: *Input*: Training i-vectors $\mathcal{X}$ and their class labels $\mathcal{Y}_{\text{cls}}$ and domain labels $\mathcal{Y}_{\text{dom}}$
3: *Output*: G_{cls}, G_{dom}, D and F
4: Initialize the weights of G_{cls}, G_{dom}, D and F using the Xavier initializer
5: **foreach** epoch **do**
6: Create N mini-batches $\{\mathcal{X}_i, \mathcal{Y}_{\text{cls},i}$ and $\mathcal{Y}_{\text{dom},i}\}_{i=1}^{N}$ of size B from $\{\mathcal{X}, \mathcal{Y}_{\text{cls}}, \mathcal{Y}_{\text{dom}}\}$
7: **for** $i = 1$ to N **do**
8: **for** $j = 1$ to B **do**
9: Compute $\hat{\mathrm{z}}_{ij} = [\mathbf{W}_1 \ \mathbf{W}_2]^{\mathsf{T}}[G_{\text{cls}}(\mathbf{x}_{ij}) \ G_{\text{dom}}(\mathbf{x}_{ij})]$, where $\mathbf{x}_{ij} \in \mathcal{X}_i$
10: **end for**
11: Train domain discriminator D using $\{\hat{\mathbf{z}}_{ij}\}_{j=1}^{B}$ as input and $\{\mathbf{y}_{\text{dom},ij}\}_{j=1}^{B}$ as target outputs of domain discriminator D, where $\mathbf{y}_{\text{dom},ij} = [y_{\text{dom},ij}^{(1)} \dots y_{\text{dom},ij}^{(M)}]^{\mathsf{T}} \in \mathcal{Y}_{\text{dom},i}$ (Eq. 4a)
12: Train adversarially domain suppressor G_{dom} using $\{\mathbf{x}_{ij}\}_{j=1}^{B}$ as input, and $[\frac{1}{M} \dots \frac{1}{M}]^{\mathsf{T}}$ as the target output of domain discriminator D, where M is the number of domains (Eq. 4b)
13: **for** $r = 1$ to R **do**
14: Train class encoder G_{cls} using $\{\mathbf{x}_{ij}\}_{j=1}^{B}$ as input, and $\{\mathbf{y}_{\text{cls},ij}\}_{j=1}^{B}$ as output of class encoder G_{cls}, where $\mathbf{y}_{\text{cls},ij} = [y_{\text{cls},ij}^{(1)} \dots y_{\text{cls},ij}^{(K)}]^{\mathsf{T}} \in \mathcal{Y}_{\text{cls},i}$ and K is the number of classes (Eq. 3b)
15: **end for**
16: Train adversarially fuzzifier F using $\{\hat{\mathbf{z}}_{ij}\}_{j=1}^{B}$ as input, and $\mathbf{f}_{\text{cls},ij} = [\frac{1}{K}, \dots, \frac{1}{K}]^{\mathsf{T}}$ as the target output of fuzzifier F (Eq. 3a)
17: **end for**
18: **end foreach**
19: **end procedure**

[20] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," *arXiv preprint arXiv:1412.4446*, 2014.

[21] Q. Wang, W. Rao, S. Sun, L. Xie, E. S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," *Proc. International conference on Acoustic, Speech, and Signal Processing*, pp. 819–824, 2018.

[22] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[23] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, 2006.

[24] M. W. Mak, "Lecture notes on factor analysis and i-vectors," in *Technical Report and Lecture Note Series*, 2016. [Online]. Available: http://www.eie.polyu.edu.hk/~mwmak/papers/FA-Ivector.pdf

[25] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[26] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[27] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2017.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[29] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, 2017.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," *arXiv preprint arXiv:1612.02136*, 2016.

[33] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[34] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[35] M. McLaren, M. I. Mandasari, and D. A. van Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2012.

[36] A. Kanervisto, V. Vestman, M. Sahidullah, V. Hautamäki, and T. Kinnunen, "Effects of gender information in text-independent and text-dependent speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5360–5364.

[37] M. Mak, X. Pang, and J. Chien, "Mixture of PLDA for noise robust i-vector speaker verification," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 130–142, 2016.

[38] A. Varga and H. J. Steeneken, "Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech communication*, vol. 12, no. 3, pp. 247–251, 1993.

[39] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot *et al.*, "Promoting robustness for speaker modeling in the community: the prism evaluation set," in *Proceedings of NIST 2011 workshop*. Citeseer, 2011.

[40] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer, Speech and Language*, vol. 28, no. 1, pp. 295–313, Jan 2014.

[41] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on ASSP*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[42] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, Crete, Greece, Jun. 2001, pp. 213–218.

[43] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, Jan. 2000.

[44] A. O. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for svm-based speaker recognition," in *Proc. Ninth International Conference on Spoken Language Processing*, 2006, p. 1471–1474.