Multi-source I-vectors Domain Adaptation using Maximum Mean Discrepancy Based Autoencoders

Weiwei Lin, Man-Wai Mak, *Senior Member, IEEE,* and Jen-Tzung Chien, *Senior Member, IEEE*

***Abstract*—Like many machine learning tasks, the performance of speaker verification (SV) systems degrades when training and test data come from very different distributions. What's more, both training and test data themselves could be composed of heterogeneous subsets. These multi-source mismatches are detrimental to SV performance. This paper proposes incorporating maximum mean discrepancy (MMD) into the loss function of autoencoders to reduce these mismatches. MMD is a nonparametric method for measuring the distance between two probability distributions. With a properly chosen kernel, MMD can match up to infinite moments of data distributions. We generalize MMD to measure the discrepancies of multiple distributions. We call the generalized MMD domain-wise MMD. Using domain-wise MMD as an objective function, we propose two autoencoders, namely nuisance-attribute autoencoder (NAE) and domain-invariant autoencoder (DAE), for multi-source i-vector adaptation. NAE encodes the features that cause most of the multi-source mismatch measured by domain-wise MMD. DAE directly encodes the features that minimize the multi-source mismatch. Using these MMD-based autoencoders as a preprocessing step for PLDA training, we achieve a relative improvement of 19.2% EER on the NIST 2016 SRE compared to PLDA without adaptation. We also found that MMD-based autoencoders are more robust to unseen domains. In the domain robustness experiments, MMD-based autoencoders show 6.8% and 5.2% improvements over IDVC on female and male Cantonese speakers, respectively.**

***Index Terms*—speaker verification, domain adaptation, i-vectors, maximum mean discrepancy.**

I. Introduction

Using i-vector as an unsupervised feature extraction method and PLDA as a supervised channel compensation technique have been very successful in speaker verification [1], [2]. However, like many machine learning algorithms, i-vector/PLDA assumes that the training data and test data are independently sampled from the same distribution. When training data and test data have a severe mismatch, the performance degrades rapidly [3]–[9]. The mismatch between training data and test data is not uncommon, as it can be caused by a lot of factors such as languages, channels, noises, and genders. Basically, collecting more data to retrain the system is time-consuming and computationally-expensive. Such a solution is unrealistic in some scenarios. It is desirable to use the existing data and a small amount of target-specific data to modify the system to meet the need, which is essentially what domain adaptation (DA) does.

Garcia-Romero and McCree [3] found that the mismatch between the out-of-domain PLDA model and the in-domain test data contributes to most of the performance loss. Therefore, it is important to apply domain adaptation to reduce the mismatch between in-domain and out-domain i-vectors before training the PLDA model. Alternatively, a PLDA model trained on out-of-domain data can be adapted to fit the in-domain data.

Earlier attempts in i-vector based DA require the in-domain data to have speaker labels. For example, Garcia-Romero and McCree [3] computed the MAP-estimates of the in-domain within-speaker and across-speaker covariance matrices in the i-vector space using the speaker labels from the in-domain data. In [5], these matrices are treated as latent variables and their joint posterior distribution is factorized using variational Bayes so that MAP point estimates of the matrices can be computed from the factorized distributions. The point estimates are then used for scoring in the target environment. More recent approaches attempt to obviate the requirement of speaker labels. For instance, Villalba and Lleida [10] extended their Bayesian adaptation in [5] by treating the *unknown* speaker labels in the in-domain data as latent variables. Borgstrom *et al.* [11] obviated the speaker-label requirement by assuming that all in-domain data in Villalba's Bayesian adaptation are independent. Another approach is to generate *hypothesized* speaker labels via unsupervised clustering [4], [12], [13]. Given the hypothesized labels, the covariance matrices of in-domain data can be computed as usual and can be interpolated with the out-of-domain covariance matrices to obtain an adapted PLDA model. Of course, correctly inferring all of the missing labels is even harder than performing speaker verification. However, as is shown in [4], even imperfect labels can achieve performance almost as good as the correct labels. Still, cluster-based approaches require a lot of heuristics to set the number of clusters.

It is also possible to carry out the unsupervised DA without inferring the missing labels at all. Most of the methods in this category assume that there is a common feature space in which in-domain and out-domain have a minimum mismatch. DA aims to project data on such feature space and uses the projected data to train a classifer. Fig 1 shows the process of i-vector based domain adaptation using the common feature-space approach. As mismatch can be caused by multiple sources, it is helpful to divide the training data into homogenous subsets according to their sources before finding a common feature space. This is called multi-source domain adaptation in the literature [14]. In addition to the robustness to heterogeneous sources, this approach also has the potential to generalize to unseen domains, as it does not assume a particular in-domain environment.

This works was in part supported by the RGC of Hong Kong SAR, Grant No. PolyU 152137/17E and PolyU 152518/16E, and the Taiwan MOST with Grant 107-2634-F-009-003.

Over the years, several unsupervised DA techniques have been proposed to find a common feature space that is less domain dependent. These techniques include inter-dataset variability compensation (IDVC) [6], [15], [16], source normalization (SN) [7] and discriminative multi-domain PLDA [9]. IDVC divides the training data into several subsets, and for each subset, the mean is computed. The means of these subsets are used to find the directions of maximum inter-dataset variability; then the subspace corresponding to these directions is removed from all i-vectors. In [6], the author successfully used IDVC to reduce the mismatch between NIST telephone data and switchboard data. In several NIST 2016 submissions [17], [18], IDVC is also found to be very helpful in boosting system performance. Recently, domain-adversarial training of neural networks has also been successfully applied to unsupervised domain adaptation for speaker recognition [19]. In domain-adversarial learning [20], a feature extraction network is trained to produce embedded features that are indistinguishable to a domain classifier but are highly speaker discriminative to a speaker classifier. After training, the embedded features are believed to be domain-invariant.

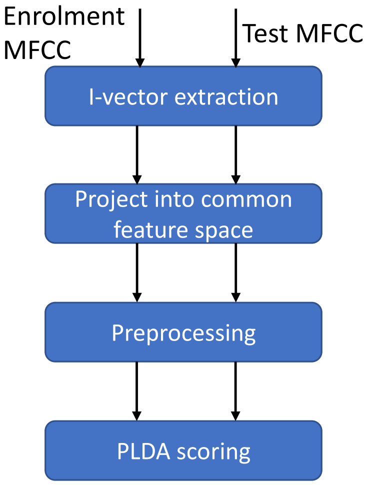

Fig. 1. A flow chart showing the process of i-vector based domain adaptation using the common feature-space approach.

Several theoretical works in DA [21]–[23] and practical applications [24] suggest that minimizing the divergence between the in-domain and out-domain distributions is very important for obtaining a good representation for DA. From this perspective, approaches based solely on the differences among the domain-means, such as IDVC, are not enough for finding a good representation. The reason is that even if the means of the distributions are exactly the same, there could still be severe mismatch between the data distributions if their variances are very different. Thus, to reduce inter-dataset mismatch, it is important to consider the statistics beyond the means.

To better utilize the statistics of multi-source data, we extend our earlier work [25] on using maximum mean discrepancy (MMD) for domain adaptation. MMD is a nonparametric method for measuring the distance between two distributions [26]–[28]. With a properly chosen kernel, MMD can utilize all moments of data. In this paper, we generalize MMD to measure the discrepancies among multiple distributions. By formulating MMD as a part of the objective function for training autoencoders, the autoencoders will learn the features that contain less domain-specific information but are still relevant to the classification task. We also apply MMD to force an autoencoder to capture the inter-data set variabilities. By subtracting out these variabilities, the i-vectors become more domain independent.

II. The I-vector/PLDA Framework

Since its first appearance [1], i-vector has become the de facto choice for the representation of utterances in speaker verification and other related areas. The i-vector approach is essentially a factor analysis technique trying to find a low-dimensional subspace that captures most of the variations in the GMM-supervectors [29]. Specifically, the GMM-supervector of utterance t can be generated by the following generative model:

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

where $\boldsymbol{\mu}$ is a supervector formed by stacking the means of a universal background model (UBM) and $\mathbf{T}$ is a low-rank total variability matrix. The posterior mean of $\mathbf{w}_t$ is the i-vector $\mathbf{x}_t$ of utterance t.

As i-vectors contain all sort of variabilities in utterances, channel compensation techniques are essential for suppressing the non-speaker variability. Among them, probabilistic discriminant analysis (PLDA) [2] performs the best. Given a set of D-dimensional length-normalized [30] i-vectors $\{\mathbf{x}_{ij}; i = 1, \ldots, N; j = 1, \ldots, H_i;\}$ from N speakers, each with H_i sessions, PLDA assumes that the i-vectors can be expressed as the following factor analysis model:

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

where $\mathbf{m}$ is the global mean of the i-vectors, $\mathbf{V}$ defines the speaker subspace, $\mathbf{z}_i$ is the speaker factor and $\boldsymbol{\epsilon}_{ij}$ is the residual noise.

III. Maximum Mean Discrepancy Autoencoder

In this section, we first highlight the domain mismatches in NIST 2016 SRE data and the limitation of IDVC. Then, we explain why maximum mean discrepancy is theoretically better than IDVC and how it can be incorporated into the training of autoencoders for extracting domain-invariant features.

A. Multi-source Mismatch in NIST 2016 SRE

NIST 2016 speaker recognition evaluation (SRE16) introduces various new challenges to speaker recognition [31], [32], among which the multilingual setup brought the most attention. Unlike previous SREs, both development (Dev) and evaluation (Eval) data in SRE16 comprise utterances spoken in non-English languages and were recorded outside north America. Table I shows the composition of SRE16 data. Because all of the SRE16 data are non-English, training using data from previous SREs results in poor performance. The mismatch

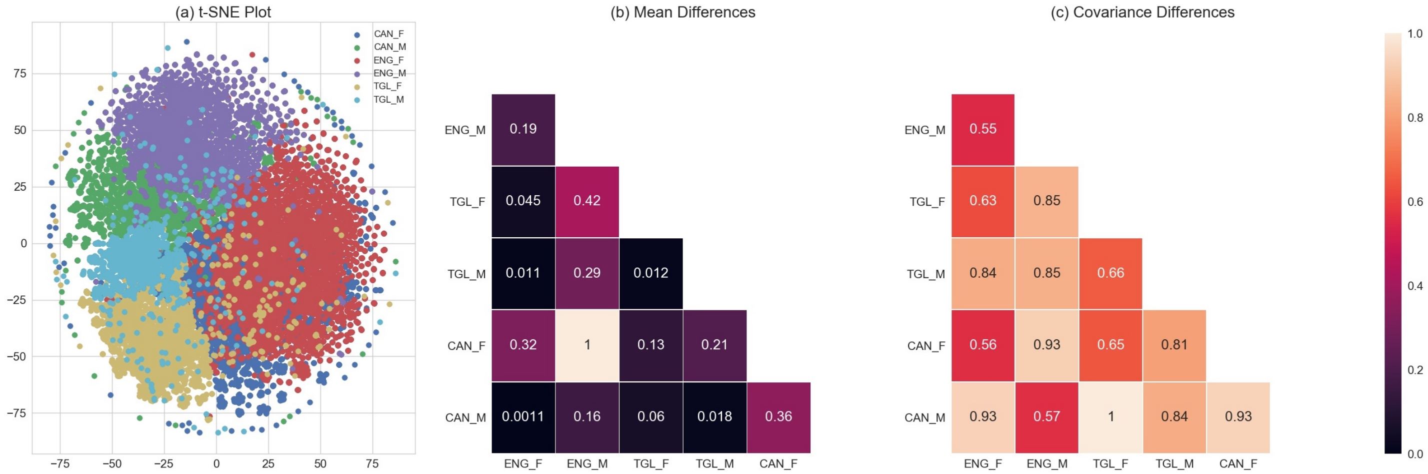

Fig. 2. (a) Scatter plot of 2-dimensional t-SNE embedded i-vectors. In the legend, “M” and “F” stand for male and female, respectively, and “ENG”, “CAN” and “TGL” stand for English, Cantonese, and Tagalog, respectively. (b) Pairwise differences between the means. (c) Pairwise differences between the covariance matrices. The means and covariances differences are measured in the *original* space and are normalized to the range between 0 and 1 for ease of comparison.

between the recording environments of previous SREs and SRE16 also causes performance degradation. Training using only SRE16 development data is also not feasible, as there are only 2,472 segments in total and a very small number of them are labeled. Besides, the labeled development data have different languages than the evaluation data.

Dataset	Category	Language
Dev	Unlabelled	Cantonese and Tagalog
Dev	Unlabelled	Mandarin and Cebuano
Dev	Labelled	Mandarin and Cebuano
Eval	Enrollment	Cantonese and Tagalog
Eval	Test	Cantonese and Tagalog

TABLE I
THE COMPOSITION OF SRE16 DATA. “LABELED” MEANS SPEAKER LABELS ARE PROVIDED. “UNLABELED” MEANS SPEAKER LABELS ARE NOT PROVIDED.

Fig. 2(a) shows the t-distributed stochastic neighbor embedding (t-SNE) [33] of i-vectors from SRE16 development data and previous SRE data. In the figure, datasets are colored according to their genders and languages. The gender- and language-dependent clusters are clearly visible in this 2-dimensional embedded space. Also, the multi-source mismatches occur not only between the English data (ENG_F and ENG_M) and SRE16 data but also within SRE16 data (CAN_F, CAN_M, TGL_F and TGL_M).

To compare the closeness between different language- and gender-clusters in the original i-vector space, we computed the squared Euclidean distances between the means of these clusters and normalized the pairwise distances by their maximum. Fig. 2(b) shows these normalized distances. Apparently, some gender-language combinations (e.g., English-male) are more distinct from the others. To get a sense of the degree of dispersion of these clusters, we may compare the maximum pairwise distance (= 0.0021) with the trace of the total covariance matrix (= 0.0068). This means that the maximum distance (occurs between CAN_F and ENG_M) is about 31% of the total variance, which is not a small value because if the i-vectors are domain-independent, this value should be zero.

To compare the variances of these clusters, Fig. 2(c) shows the pairwise differences between the covariance matrices of the i-vectors from the six language- and gender-dependent groups. The differences are measured in terms of Frobenius norm [34]. Again, the differences are normalized by the maximum difference. We can see that the covariance matrices of these clusters are fairly different from each other. In particular, even the minimum difference (ENG_M–ENG_F) is 55% of the maximum difference (CAN_M–TGL_F). Interestingly, although CAN_M and ENG_F are closest in terms of their means, the difference between their covariance matrix is the second largest. Overall speaking, Fig. 2 shows that the i-vectors of these six groups differ from each other not only by their means but also by their covariance matrices.

B. Inter-dataset Variability Compensation

Inter-dataset variability compensation (IDVC) was proposed in [6]. IDVC follows the subspace removal approach proposed in [35]. It aims to find the directions in the i-vector space with the largest inter-dataset variability and remove the variability in these directions. This is achieved by projecting the i-vectors $\mathbf{x}$'s as follows:

$$\hat{\mathbf{x}} = (\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathsf{T}})\mathbf{x}, \tag{3}$$

where the columns of $\mathbf{W}$ span the subspace of unwanted variability. $\mathbf{W}$ comprises the eigenvectors of the covariance matrix of the subset means. The idea of IDVC has been extended to compensating for the variability in the PLDA hyperparameters (between- and within-speaker covariance matrices) due to domain mismatch [16]. Recently, it has been extended to reduce the inaccuracy of PLDA scores caused by domain mismatch [15].

Note that in IDVC the domain mismatch is defined by the covariances of subset means. However, the mismatch of datasets may not only manifest in the dataset means, but also in the higher-order statistics of these datasets. The limitation of IDVC will become apparent when we consider some Gaussian distributions (one for each dataset) with identical means but different covariance matrices. Despite of the severe mismatches among these Gaussians, IDVC considers these Gaussians to be identical and will not remove any subspace ($\mathbf{W}$ in Eq. 3 is a null matrix) to reduce the mismatches.

C. Maximum Mean Discrepancy

The theoretical works in DA [21]–[23] suggest that it is important to have a good measurement of the divergence between the data distributions of different domains. Maximum mean discrepancy is a distance measure in the space of probability. Given two sets of samples $\{\mathbf{x}_i\}_{i=1}^N$ and $\{\mathbf{y}_j\}_{j=1}^M$, MMD computes the mean squared difference of the statistics of the two datasets:

$$\mathcal{D}_{\text{MMD}} = \left\| \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x}_i) - \frac{1}{M}\sum_{j=1}^{M} \phi(\mathbf{y}_j) \right\|^2, \tag{4}$$

where ϕ is a feature map. When ϕ is the identity function, the MMD distance simply computes the discrepancy between the sample means.

Eq. 4 can be expanded as:

$$\begin{aligned}\mathcal{D}_{\text{MMD}} = & \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} \phi(\mathbf{x}_i)^{\mathsf{T}}\phi(\mathbf{x}_{i'}) \\ & - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \phi(\mathbf{x}_i)^{\mathsf{T}}\phi(\mathbf{y}_j) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} \phi(\mathbf{y}_j)^{\mathsf{T}}\phi(\mathbf{y}_{j'}).\end{aligned} \tag{5}$$

As each term in Eq. 5 involves dot products only, the kernel trick can be applied:

$$\begin{aligned}\mathcal{D}_{\text{MMD}} = & \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i'=1}^{N} k(\mathbf{x}_i, \mathbf{x}_{i'}) \\ & - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(\mathbf{x}_i, \mathbf{y}_j) + \frac{1}{M^2}\sum_{j=1}^{M}\sum_{j'=1}^{M} k(\mathbf{y}_j, \mathbf{y}_{j'}),\end{aligned} \tag{6}$$

where $k(\cdot,\cdot)$ is a kernel function. In the case of quadratic (Quad) kernels, we have:

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathsf{T}}\mathbf{y} + c)^2. \tag{7}$$

Then, the MMD becomes:

$$\begin{aligned}\mathcal{D}_{\text{MMD}} = 2c & \left\| \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i - \frac{1}{M}\sum_{j=1}^{M} \mathbf{y}_j \right\|^2 \\ & + \left\| \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^{\mathsf{T}} - \frac{1}{M}\sum_{j=1}^{M} \mathbf{y}_j\mathbf{y}_j^{\mathsf{T}} \right\|_F^2,\end{aligned} \tag{8}$$

where $\|\cdot\|_F$ represents the Frobenius norm. We can see that with a quadratic kernel, MMD can match up to the second-order statistics and c can be adjusted to control the trade-off of the matching between the first-order and the second-order moments.

Another popular kernel is the radial basis function (RBF) kernel:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{y}\|^2 \right), \tag{9}$$

where σ is the width parameter. With the RBF kernel, the feature space is of infinite dimension and contains all moments of data. Minimizing MMD using the RBF kernel is equivalent to matching all moments of two distributions [27]. It is also possible to use a mixture of RBF kernels [28]:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{q=1}^{K} \exp\left(-\frac{1}{2\sigma_q^2} \|\mathbf{x} - \mathbf{y}\|^2 \right), \tag{10}$$

where σ_q is the width parameter of the q-th RBF kernel.

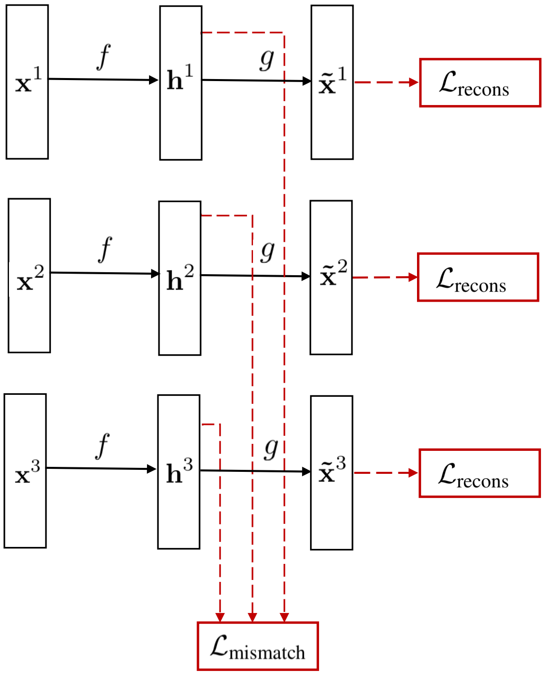

Fig. 3. Architecture of the proposed domain-invariant autoencoder (DAE) when data are from three different domains. Solid black arrows represent the connections between neurons. Dashed red arrows represent the hidden nodes' outputs for computing the domain-mismatch loss or autoencoder's outputs for computing the reconstruction loss.

D. Domain-invariant Autoencoder

Assume that we have in-domain data $\{\mathbf{x}_i^{\text{in}}\}_{i=1}^{N_{\text{in}}}$ and out-domain data $\{\mathbf{x}_j^{\text{out}}\}_{j=1}^{N_{\text{out}}}$. We want to learn a transform $\mathbf{h} = f(\mathbf{x})$ such that the transformed data $\{\mathbf{h}_i^{\text{in}}\}_{i=1}^{N_{\text{in}}}$ and $\{\mathbf{h}_j^{\text{out}}\}_{j=1}^{N_{\text{out}}}$ are as similar as possible. The mismatch between the transformed data can be measured by MMD:

$$\begin{aligned}\mathcal{D}_{\text{MMD}} = & \frac{1}{N_{\text{in}}^2}\sum_{i=1}^{N_{\text{in}}}\sum_{i'=1}^{N_{\text{in}}} k(\mathbf{h}_i^{\text{in}}, \mathbf{h}_{i'}^{\text{in}}) \\ & - \frac{2}{N_{\text{in}}N_{\text{out}}}\sum_{i=1}^{N_{\text{in}}}\sum_{j=1}^{N_{\text{out}}} k(\mathbf{h}_i^{\text{in}}, \mathbf{h}_j^{\text{out}}) + \frac{1}{N_{\text{out}}^2}\sum_{j=1}^{N_{\text{out}}}\sum_{j'=1}^{N_{\text{out}}} k(\mathbf{h}_j^{\text{out}}, \mathbf{h}_{j'}^{\text{out}}).\end{aligned} \tag{11}$$

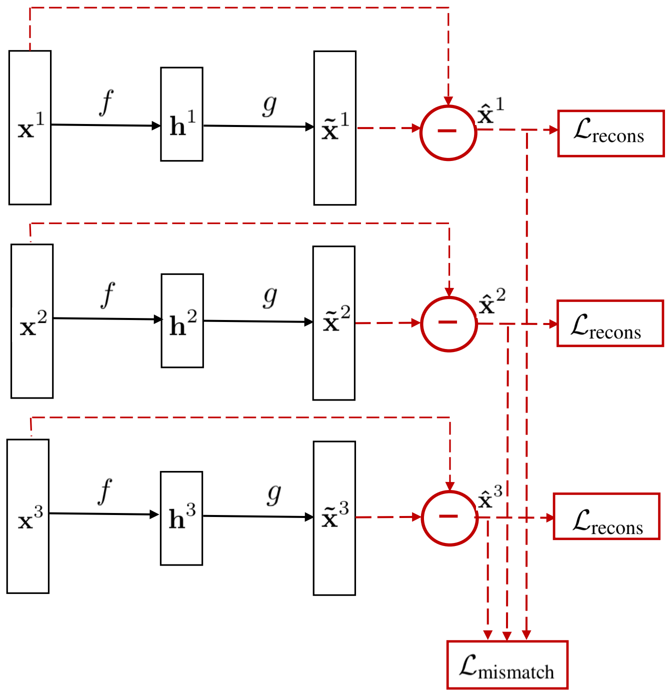

Fig. 4. Architecture of the proposed nuisance-attribute autoencoder (NAE) when data are from three different domains. Solid black arrows represent the connections between neurons. Dashed red arrows represent the signal pathways for computing the domain-mismatch loss or reconstruction loss.

When the data come from multiple sources, we want the transformed data to be as similar to each other as possible. To this end, we define a domain-wise MMD measure. Specifically, given D sets of data $\{\mathbf{x}_i^d\}_{i=1}^{N_d}$, where $d = 1, 2, \ldots, D$, we want the transformed data $\{\mathbf{h}_i^d\}_{i=1}^{N_d}$ to have small loss as defined by the following equation:

$$\mathcal{L}_{\text{mismatch}} = \sum_{d=1}^{D}\sum_{\substack{d'=1\\ d'\neq d}}^{D}\left(\frac{1}{N_d^2}\sum_{i=1}^{N_d}\sum_{i'=1}^{N_d}k(\mathbf{h}_i^d,\mathbf{h}_{i'}^d) - \frac{2}{N_dN_{d'}}\sum_{i=1}^{N_d}\sum_{j=1}^{N_{d'}}k(\mathbf{h}_i^d,\mathbf{h}_j^{d'}) + \frac{1}{N_{d'}^2}\sum_{j=1}^{N_{d'}}\sum_{j'=1}^{N_{d'}}k(\mathbf{h}_j^{d'},\mathbf{h}_{j'}^{d'})\right). \tag{12}$$

Of course, we also want to retain as much non-domain related information as possible. Assume that another transform can reconstruct the input from $\mathbf{h}$:

$$\tilde{\mathbf{x}} = g(\mathbf{h}), \tag{13}$$

where $\tilde{\mathbf{x}}$ is the reconstruction of the input $\mathbf{x}$. We want to make $\tilde{\mathbf{x}}$ as close to $\mathbf{x}$ as possible by minimizing:

$$\mathcal{L}_{\text{recons}} = \frac{1}{2}\sum_{d=1}^{D}\sum_{i=1}^{N_d}\left\|\mathbf{x}_i^d - \tilde{\mathbf{x}}_i^d\right\|^2. \tag{14}$$

Both objectives can be achieved by an antoencoder comprising an encoder network f and a decoder network g, with the total loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mismatch}} + \lambda\mathcal{L}_{\text{recons}}, \tag{15}$$

where λ is a parameter controlling the importance of the reconstruction loss. Note that both f and g can be multilayer neural networks. As the autoencoder encodes domain-invariant information, we call it **d**omain-invariant **a**uto**e**ncoder (DAE).[1]

Fig. 3 shows the architecture of DAE for the case of three domains ($D = 3$), with each row corresponding to one domain. Note that the weights in the rows are shared across all domains. Fig. 5 shows a scatter plot of 2-dimensional t-SNE embedded of the hidden activations of a DAE. Compared with the t-SNE plot in Fig 2(a), we can see that the embedding of the hidden activations have apparently less domain-clustering effect than the embedding of i-vectors, which shows that the DAE indeed learns a domain-invariant representation.

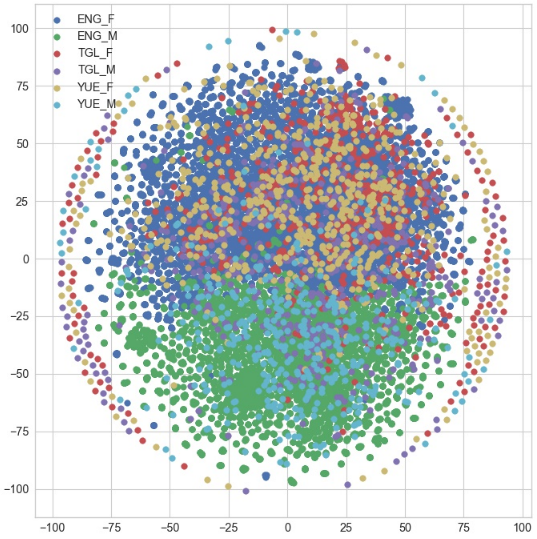

Fig. 5. Scatter plot of 2-dimensional t-SNE embedding of the hidden activations of DAE. In the legend, "M" and "F" stand for male and female, respectively, and "ENG", "CAN" and "TGL" stand for English, Cantonese, and Tagalog, respectively.

E. Nuisance-attribute Autoencoder

In addition to directly learning domain-invariant features, we can also train an autoencoder to remove the domain-specific features similar to the idea of IDVC. Note that Eq. 3 can be written as:

$$\hat{\mathbf{x}} = \mathbf{x} - \mathbf{W}\mathbf{W}^{\mathsf{T}}\mathbf{x}, \tag{16}$$

where $\mathbf{W}\mathbf{W}^{\mathsf{T}}\mathbf{x}$ can be interpreted as the nuisance components. In stead of using principal component analysis (PCA) as in IDVC, we may use an autoencoder to estimate the nuisance components. Specifically, Eq. 16 can be replaced by:

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{x} - \tilde{\mathbf{x}}\\ &= \mathbf{x} - g(f(\mathbf{x})),\end{aligned} \tag{17}$$

where $g(f(\mathbf{x}))$ is implemented by a special form of autoencoders. Similar to the DAE, the autoencoder has an encoder $\mathbf{h} = f(\mathbf{x})$ and a decoder $\tilde{\mathbf{x}} = g(\mathbf{h})$. But unlike the DAE, the autoenconder is trained to make $\hat{\mathbf{x}}$ as close to $\mathbf{x}$ as possible.

[1] Not to be confused with the denoising autoencoder of Vincent *et al* [36].

The subtraction in Eq. 17 will make $\tilde{\mathbf{x}}$'s to contain all of the domain-specific information and $\hat{\mathbf{x}}$ to become less domain-dependent.

To achieve the goal mentioned above, we can use MMD to measure the discrepancy between the distribution of $\hat{\mathbf{x}}$ across different datasets:

$$\mathcal{L}_{\text{mismatch}} = \sum_{d=1}^{D} \sum_{\substack{d'=1 \\ d' \neq d}}^{D} \Bigg(\frac{1}{N_d^2} \sum_{i=1}^{N_d} \sum_{i'=1}^{N_d} k(\hat{\mathbf{x}}_i^d, \hat{\mathbf{x}}_{i'}^d) - \frac{2}{N_d N_{d'}} \sum_{i=1}^{N_d} \sum_{j=1}^{N_{d'}} k(\hat{\mathbf{x}}_i^d, \hat{\mathbf{x}}_j^{d'}) + \frac{1}{N_{d'}^2} \sum_{j=1}^{N_{d'}} \sum_{j'=1}^{N_{d'}} k(\hat{\mathbf{x}}_j^{d'}, \hat{\mathbf{x}}_j^{d'}) \Bigg). \quad (18)$$

Also, we can add reconstruction loss between $\mathbf{x}$ and $\hat{\mathbf{x}}$. As this network encodes the unwanted domain-specific information, we call it **n**uisance-attribute **a**uto**e**ncoder (NAE). Fig. 4 shows the architecture of NAE for the case of three domains ($D = 3$).

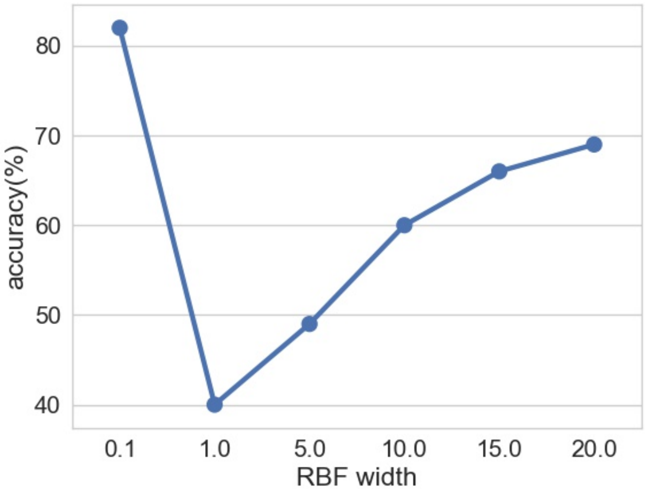

Fig. 6. The relationship between the width σ of the radial basis function kernel and the accuracy of the softmax domain classifier on the features extracted from a DAE.

IV. Experimental Setup

A. Speech Data and Acoustic Features

Speech files from NIST 2004–2010 Speaker Recognition Evaluation (hereafter, referred to as SRE04–SRE10)[2] and the development set of SRE16 (SRE16-dev) were used as development data and speech files from the evaluation set of SRE16 (SRE16-eval) were used as test data. The speech regions in the speech files were extracted by using a two-channel voice activity detector [37]. For each speech frame, 19 MFCCs together with energy plus their 1st and 2nd derivatives were computed, followed by cepstral mean normalization and feature warping [38] 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 and PLDA Model Training

The i-vector/PLDA system is based on a gender-independent UBM with 512 mixtures and a gender-independent total variability matrix with 300 total factors. Unlabelled and enrollment utterances from SRE16 development data were used for training the UBM and the total variability (TV) matrix. The TV matrix and UBM were used for extracting i-vectors from the speech files (both gender) in SRE04–SRE10, SRE16–dev and SRE16–eval.

Unless stated otherwise, i-vectors derived from SRE04–SRE10 and the SRE16 development set were used for training the DAEs, the NAEs, and the projection matrices of IDVC. Note that the non-English speech in SRE04–SRE10 were filtered out. The resulting networks and projection matrices were then used to transform i-vectors derived from SRE04–SRE10 and SRE16. Then, we computed a PCA projection matrix by using the transformed i-vectors from SRE04–SRE10 and reduced the dimension of i-vectors to 200. Length normalization were applied to the 200-dimensional i-vectors [30]. Then, we trained a gender-independent PLDA model with 200 latent variables using SRE04–SRE10 data only. PLDA scores were normalized by S-norm using SRE16 development data as the cohort set [39].

Speech files with bad recordings (e.g., without speech or contain telephone tones only) as detected by the VAD and speech files shorter than 10 seconds were excluded from training any models (UBM, TV matrix, and PLDA) and networks (DAEs and NAEs).

C. MMD Autoencoders and IDVC Training Details

We used quadratic kernels and RBF kernels for MMD, and used a softmax (multi-class logistic regression) domain classifier to determine the best hyperparameters of the RBF kernel. Specifically, for each candidate RBF width, we trained a DAE and extracted vectors from the hidden nodes' activations in Fig. 3 and used them as the input to a softmax classifier with the number of outputs equals to the number of domains. Similarly, another softmax domain classifier was trained to classify the domain-removed vectors $\hat{\mathbf{x}}$ in Fig. 4. Because our goal is to make these vectors as domain-invariant as possible, we selected the RBF width such that the resulting MMD vectors and domain-removed vectors lead to the lowest accuracy in the domain classifiers.

Fig. 6 shows the classification accuracy of the MMD vectors $\mathbf{h}$'s. with respect to the RBF width. Evidently, the MMD vectors contain the least domain information when $\sigma = 1$, as they lead to the lowest domain classification accuracy. For both DAE and NAE, the weights in the decoder and encoder networks are always tied as in [40]. Unless stated otherwise, we divided SRE04–10 and SRE16 data into gender- and language-homogenous subsets to train the IDVC projection matrices, the DAEs, and the NAEs. Excluding the minor data in SRE16, we have six subsets: English male, English female, Cantonese male, Cantonese female, Tagalog male and Tagalog female. The networks were optimized using the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [41], [42]. The history size of L-BFGS was

[2] https://www.nist.gov/itl/iad/mig/speaker-recognition

	Architecture	EER (%)	mCprim	aCprim	$\mathcal{L}_{\text{mismatch}}$	$\mathcal{L}_{\text{recons}}$	$\mathcal{L}_{\text{total}}$
No Adapt	-	15.84	0.89	0.93	-	-	-
PCA	-	14.77	0.89	0.92	-	-	-
IDVC	-	13.08	0.86	0.93	-	-	-
Linear DAE	300-300(linear)-300	**12.79**	0.85	0.91	0.002	0.012	0.014
Non-linear DAE	300-300(sigm)-300(linear)-300(linear)-300(sigm)-300	24.36	0.98	0.99	0.001	0.220	0.221
Linear NAE	300-10(linear)-300	12.81	0.85	0.91	0.003	0.211	0.214
Non-linear NAE	300-10(sigm)-10(linear)-10(linear)-10(sigm)-300	14.73	0.93	0.93	0.156	2.032	2.188

TABLE II
The Performance of four domain adaptation methods and the performance of a classical i-vector PLDA system without domain adaptation (*No Adapt*) in the SRE16 evaluation set. “Linear” and “sigm” mean that the hidden nodes in the DAE and NAE use linear and sigmoid activation functions, respectively. *PCA*: The weights of the linear autoencoder were found by PCA. *IDVC*: Inter-dataset variability compensation. “mCprim” and “aCprim” are the minimum detection cost and the actual detection cost as specified in the evaluation plan of SRE16.

set to 20 and the learning rate was set to 1. Training was stopped when the difference of loss between two iterations was smaller than 0.0001.

V. Results and discussions

A. General Performance Analysis

Table II shows the performance of four i-vector adaptation methods. All systems use PLDA as their backend. A classical i-vector PLDA system without domain adaptation (No Adapt) is also included for comparison. For the DAEs and NAEs, we used a quadratic kernel with $c = 1$ and λ in Eq. 15 was set to 1. Both linear and non-linear autoencoders were used in the experiments.

We can also see from Table II that except for non-linear DAE, all of the DA methods boost the performance significantly in term of EER, although in terms of minimum Cprimary and actual Cprimary, the improvement is minor. We can also observe that the linear DAE and NAE have a small improvement over IDVC. Surprisingly, the non-linear DAE and NAE perform worse than their linear counterparts. When we look into the losses of these autoencoders, we found that the non-linear DAE and NAE produce higher loss than their linear counterparts. Considering that non-linear autoencoders should have higher capacity in fitting the training data (but they fail to do so), they probably got stuck in local minima [43]–[45]. Because of the relatively poor performance of non-linear autoencoders, we only present and discuss the results of linear autoencoders in the sequel.

In [46], the authors showed that a linear autoencoder works like PCA if their weights are found by minimizing the mean-squared error (MSE). A natural question which arises is whether the linear DAE and NAE will reduce to PCA. Note that in [46], the objective function of the autoencoders comprises the MSE loss only. On the other hand, the objective function of DAE and NAE comprises both the MMD loss and MSE loss. As MMD loss is totally different from MSE loss, DAEs and NAEs will not reduce to PCA even with linear units. To demonstrate that our linear NAE and DAE are different from PCA, we also report results obtained by using PCA to preprocess the i-vectors in Table II. Evidently, using PCA alone could not improve the performance significantly.

To gain more insights into the performance of IDVC, DAEs, and NAEs, we report the performance of the three systems on four gender- and language-dependent subsets in Table III(a) and Fig 7. The results suggest that Tagalog is more challenging than Cantonese, with EERs of 20.55% and 19.89% for male and female, respectively. Also, the female subsets seem to be more difficult than the male ones. The performance of the four subsets improves significantly after applying the three domain adaptation methods.

B. Robustness to Unseen Domains

In the previous section, we partitioned the training data into gender- and language-homogenous groups. There are always data in the training set that match both the gender and the language of the test set. However, it is not always feasible to obtain training data that match the gender and language of the test data for domain adaptation. Therefore, we conducted a domain robustness experiment. Specifically, for each gender and language (Tagalog or Cantonese) in test sessions, we excluded the speech of the same gender who speak that language from training. In other words, there is no in-domain data for domain adaptation. Here, the term “domain” refers to genders and languages, and in-domain data are evaluation data with a specific combination of gender and language. For example, for the evaluation of male Tagalog, we exclude male Tagalog data for training the IDVC, DAE and NAE. Note that the gender and language information can be obtained from the key file of the development data provided by NIST.

Table III(b) shows the results of the three DA methods on unseen domains. Fig. 7 shows the EERs of the DA methods with and without using in-domain training data. Not surprisingly, without in-domain data for training, the performance of all DA methods degrades. Despite of the performance degradation, the performance of these DA methods are still better than the one without domain adaptation. More importantly, the DAE and NAE appear to suffer less when compared with IDVC. In particular, for Cantonese speakers, the DAE has 6.8% and 5.2% relative improvement over IDVC for female and male speakers, respectively. We believe that the incorporation of high moments of the data distributions is the reason that MMD-based methods are more robust to unseen domains.

	Cantonese						Tagalog					
	Female			Male			Female			Male		
	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim
No Adapt	10.92	0.77	0.87	10.87	0.74	0.96	20.55	0.93	0.94	19.89	0.94	0.96
IDVC	9.47	0.74	0.88	8.74	0.68	0.96	17.50	0.91	0.93	15.75	0.90	0.96
DAE	**9.15**	**0.73**	0.84	**8.61**	0.67	0.94	**17.26**	0.90	0.91	15.59	0.89	0.94
NAE	9.27	0.74	**0.83**	8.73	0.67	0.94	17.34	0.90	0.91	15.59	0.89	0.94

(a)

	Cantonese						Tagalog					
	Female			Male			Female			Male		
	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim
No Adapt	10.92	0.77	0.87	10.87	0.74	0.96	20.55	0.93	0.94	19.89	0.94	0.96
IDVC	10.22	0.75	0.87	9.67	0.70	0.96	18.11	0.91	0.93	**16.71**	0.92	0.95
DAE	**9.52**	**0.73**	0.83	**9.17**	**0.68**	0.95	**17.74**	0.91	**0.91**	16.83	**0.91**	0.94
NAE	9.82	0.74	**0.82**	9.44	0.69	0.95	17.93	**0.91**	0.92	16.89	0.92	0.94

(b)

TABLE III
THE PERFORMANCE OF VARIOUS DOMAIN ADAPTATION METHODS ON THE SUBSETS OF THE SRE16 EVALUATION SET. IN (A), THE IDVC, DAE AND NAE WERE TRAINED USING BOTH IN-DOMAIN DATA AND OUT-DOMAIN DATA. IN (B), THE IDVC, DAE AND NAE WERE TRAINED WITHOUT USING IN-DOMAIN DATA.

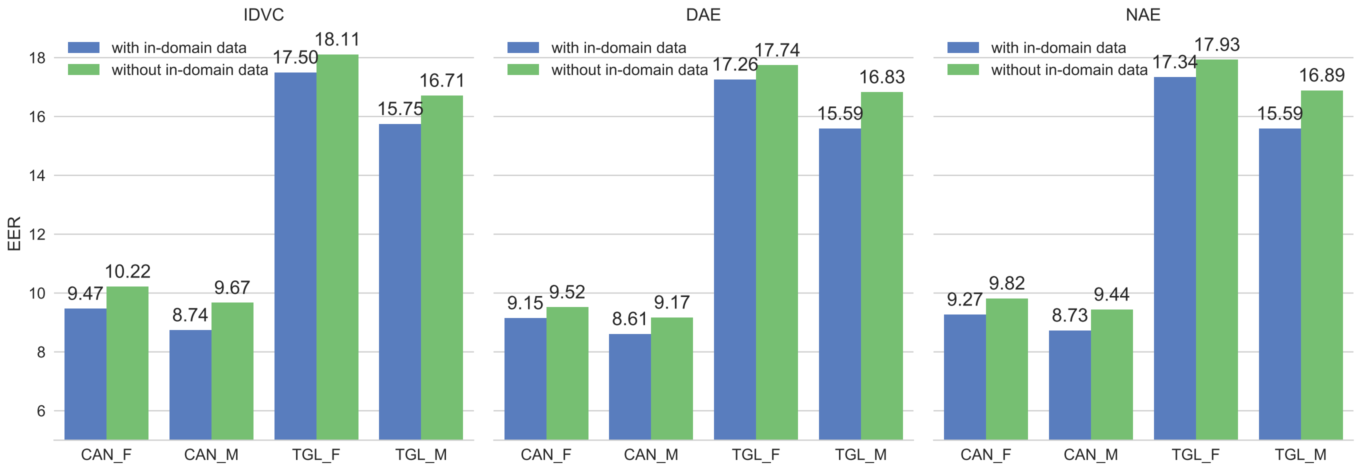

Fig. 7. Bar charts showing the EERs of three domain adaptation methods with and without using in-domain data.

C. Impact of the Hyperparameters

Comparing with IDVC, DAEs and NAEs have more hyperparameters to tune. In this subsection, we present the results of DAEs and NAEs with different values of λ and different choices of kernels. The kernels we experimented with include quadratic kernels with $c = 0$ and $c = 1$, RBF kernels with $\sigma = 1$, and the mixture of four RBF kernels with width equals to 1, 3, 5, and 10, respectively. Tables IV(a) and IV(b) show the results of DAEs and NAEs with different choices of λ's and kernels. Fig. 8 shows the EERs of DAEs and NAEs with different choices of λ and kernels.

Three phenomena can be observed from the results in Table IV and Fig. 8. Firstly, quadratic kernels and RBF kernels require different values of λ to obtain good performance. Specifically, both NAEs and DAEs with quadratic kernels perform the best when λ is equal to 0.1 or 1, while NAEs and DAEs with RBF kernels perform the best when λ is equal to 0.01 or 0.1. Secondly, for NAEs, the quadratic kernel with $c = 0$ generally performs poorly in most cases. Recall that c controls the trade-off between the matching in the first and the second moments. It seems that matching the second moments alone is not enough in most cases. Thirdly, there is no noticeable performance gain from using RBF kernels or a mixture of RBF kernels. Theoretically, RBF kernels can match up to infinite moments of data distributions and are therefore better than quadratic kernels for reducing the domain mismatch. However, in our experiments, RBF kernels have no advantage over quadratic kernels. It may be due to the fact that PLDA only uses up to the second moment.

D. Impact of Data Partition

In the previous sections, we partitioned i-vectors by both genders and languages. In this section, we investigated the influence of different partitioning schemes. According to the

	Quad(c=0)			Quad(c=1)			RBF			Mixture RBF		
λ	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim
0.01	13.76	0.86	0.93	14.07	0.87	0.91	**12.94**	0.85	0.92	**13.01**	0.85	0.90
0.1	**12.81**	**0.85**	**0.91**	12.85	**0.84**	0.90	13.29	0.85	0.90	13.23	0.85	0.91
1	12.90	0.86	0.93	**12.79**	0.85	0.91	14.13	0.87	0.89	14.05	0.88	0.91
10	13.73	0.87	0.89	13.36	0.86	0.90	14.26	0.87	0.89	14.31	0.87	0.89

(a)

	Quad(c=0)			Quad(c=1)			RBF			Mixture RBF		
λ	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim
0.01	**13.27**	0.86	0.91	12.80	0.85	0.91	**12.97**	0.85	0.92	13.11	0.85	0.91
0.1	13.47	**0.85**	0.93	**12.79**	0.85	0.91	13.02	0.85	0.90	**13.04**	0.85	0.91
1	13.72	0.86	**0.90**	12.81	0.85	0.91	13.81	0.85	**0.89**	14.10	0.85	**0.90**
10	13.80	0.86	0.91	13.05	0.85	0.91	13.99	0.85	0.92	13.90	0.88	0.91

(b)

TABLE IV

THE PERFORMANCE OF (A) DAEs AND (B) NAEs WITH DIFFERENT CHOICES OF KERNELS AND λ'S. QUAD IS THE QUADRATIC KERNEL IN EQ. 7. BOTH DAEs AND NAEs WERE TRAINED BY PARTITIONING SRE04–10 AND SRE16 DEVELOPMENT DATA INTO GENDER- AND LANGUAGE-HOMOGENOUS GROUPS.

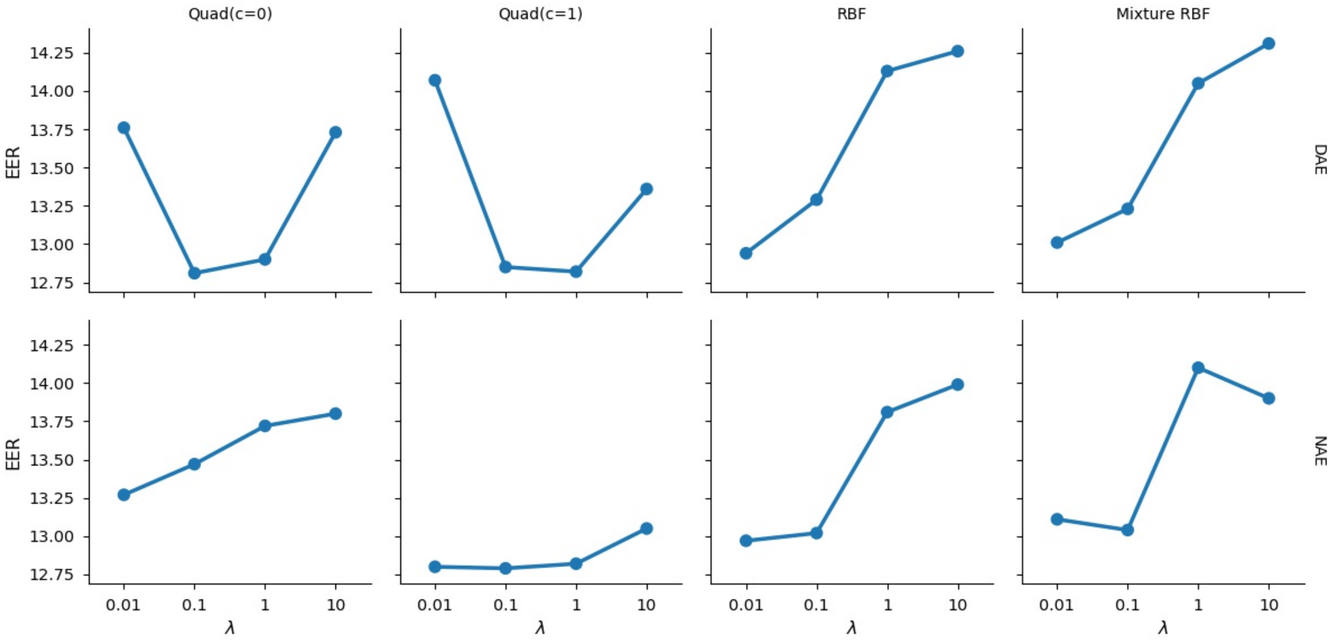

Fig. 8. Line plots showing the EERs of DAEs (top row) and NAEs (bottom row) with different choices of λ's and kernels.

Partitioning Scheme	IDVC			DAE			NAE		
	EER	mCprim	aCprim	EER	mCprim	aCprim	EER	mCprim	aCprim
Gender and Language	13.08	0.86	0.93	12.79	0.85	0.91	12.81	0.85	0.91
Gender	13.75	0.87	0.9	13.09	0.85	0.90	13.03	0.86	0.92
Language	13.59	0.85	0.93	13.29	0.85	0.90	13.29	0.85	0.90

TABLE V

THE PERFORMANCE OF IDVC, DAE AND NAE USING DIFFERENT PARTITION SCHEMES.

gender and language, we can partition data into gender-homogenous groups, language-homogenous groups or gender- and language-homogenous groups. Table V shows the results of the DA methods using different partitioning schemes. For all of the three DA methods, it is apparent that partitioning by both genders and languages achieves the best result, which clearly demonstrates the advantage of multi-source domain adaptation.

E. Combined with PLDA Model Interpolation

It is also possible to combine i-vector domain adaptation with PLDA model interpolation [4]. Specifically, the unlabelled i-vectors in the target domain were clustered to obtain some hypothesized labels, which were used to train a PLDA model. As PLDA models obtained in this way may not be very reliable, a better approach is to linearly interpolate the parameters of source as well as target PLDA models [4]. Table VI shows the results of i-vector adaptation in combination with model interpolation. The interpolation parameter was set to 0.3. When compared with using the unadapted i-vectors, it is clear that adapting the i-vectors improves performance of the PLDA model interpolation. Also, the best performance was achieved by the proposed DAE.

Adaptation Method	EER	mCprim	aCprim
No Adaptation	13.47	0.86	0.91
IDVC	12.88	0.85	0.93
DAE	12.43	0.84	0.90
NAE	12.51	0.84	0.91

TABLE VI
THE PERFORMANCE OF UNSUPERVISED PLDA MODEL INTERPOLATION USING UNADAPTED I-VECTORS AND I-VECTORS ADAPTED BY IDVC, DAES AND NAES. THE INTERPOLATION PARAMETER WAS SET TO 0.3

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two MMD-based autoencoders for multiple-source i-vector domain adaptation. Unlike IDVC, the domain-wise MMD can utilize second, third and even infinite moments of data distributions for measuring the domain mismatch. The experiments on SRE16 show that both autoencoders can significantly improve SV performance. The experiments also demonstrate that the proposed methods are more robust to unseen domains than IDVC.

Despite the promising results, there are still some problems we have not solved. For example, we have not fully utilized the rich representations that the non-linear autoencoders can offer. As shown in Table II, the non-linear autoencoders perform worse than the linear ones. As the non-linear autoencoders produce higher losses, they obviously stuck in local minima. There are promising results in the literature for incorporating more supervised objectives to reduce the chance of stucking in local minima in representation learning [44], [45]. In future work, we could incorporate metric learning [47]–[49] in the objective functions as a way to guide the learning of the autoencoders.

Currently, we separate the optimization of MMD kernel parameters and autoencoders' weights. As a result, the kernel parameters may not be optimal. However, how to jointly optimize the kernel parameters and network parameters is still an open problem, although some success in this direction has been seen recently [50]. Joint optimization is a very promising direction to look at in the future.

REFERENCES

[1] N. Dehak, P. 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.

[2] P. Li, Y. Fu, U. Mohammed, J. Elder, and S. Prince, "Probabilistic models for inference about identity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 144–157, 2012.

[3] 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*, pp. 4047–4051, 2014.

[4] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proc. Odyssey*, pp. 260–264, 2014.

[5] J. Villalba and E. Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Proc. Odyssey*, pp. 47–54, 2012.

[6] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4002–4006, 2014.

[7] M. McLaren and D. Van Leeuwen, "Source-normalized LDA for robust speaker recognition using i-vectors from multiple speech sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 755–766, 2012.

[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*, pp. 4032–4036, 2014.

[9] A. Sholokhov, T. Kinnunen, and S. Cumani, "Discriminative multi-domain PLDA for speaker verification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5030–5034, 2016.

[10] J. Villalba and E. Lleida, "Unsupervised adaptation of PLDA by using variational Bayes methods," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 744–748, 2014.

[11] B. J. Borgstrom, D. A. Reynolds, E. Singer, and O. Sadjadi, "Improving the effectiveness of speaker verification domain adaptation with inadequate in-domain data," tech. rep., MIT Lincoln Laboratory, Lexington, United States, 2017.

[12] 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*, pp. 265–272, 2014.

[13] 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*, pp. 5269–5273, 2018.

[14] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[15] H. Aronowitz, "Compensating inter-dataset variability in PLDA hyper-parameters for robust speaker recognition," in *Proc. Odyssey*, pp. 282–286, 2014.

[16] H. Aronowitz, "Inter dataset variability modeling for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5400–5404, 2017.

[17] K. A. Lee *et al.*, "The I4U mega fusion and collaboration for NIST speaker recognition evaluation 2016," in *Proc. Interspeech*, pp. 1328–1332, 2017.

[18] O. Plchot, P. Matějka, A. Silnova, O. Novotný, M. D. Sánchez, J. Rohdin, O. Glembek, N. Brümmer, A. Swart, J. Jorrn-Prieto, P. García, L. Buera, P. Kenny, J. Alam, and G. Bhattacharya, "Analysis and description of ABC submission to NIST SRE 2016," in *Proc. Interspeech*, pp. 1348–1352, 2017.

[19] 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 International Conference on Acoustics, Speech and Signal Processing*, pp. 4889–4893, 2018.

[20] 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.

[21] S. B. David, T. Lu, T. Luu, and D. Pál, "Impossibility theorems for domain adaptation," in *Proc. International Conference on Artificial Intelligence and Statistics*, pp. 129–136, 2010.

[22] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[23] P. Germain, A. Habrard, F. Laviolette, and E. Morvant, "A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers," in *Proc. International Conference on Machine Learning*, pp. 738–746, 2013.

[24] H.-Y. Chen and J.-T. Chien, "Deep semi-supervised learning for domain adaptation," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2015.

[25] 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*, pp. 162–167, 2018.

[26] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Advances in Neural Information Processing systems*, pp. 513–520, 2007.

[27] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. International Conference on Machine Learning*, pp. 1718–1727, 2015.

[28] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. International Conference on Machine Learning*, pp. 97–105, 2015.

[29] 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, no. 5, pp. 308–311, 2006.

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

[31] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," in *Proc. Interspeech*, pp. 1353–1357, 2017.

[32] K. Jones, S. Strassel, K. Walker, D. Graff, and J. Wright, "Call My Net corpus: A multilingual corpus for evaluation of speaker recognition technology," in *Proc. Interspeech*, pp. 2621–2624, 2017.

[33] 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.

[34] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[35] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition.," in *Proc. Odyssey*, vol. 4, pp. 219–226, 2004.

[36] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. International Conference on Machine learning*, pp. 1096–1103, ACM, 2008.

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

[38] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Odyssey*, 2001.

[39] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Interspeech*, 2017.

[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[42] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, 1989.

[43] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[44] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT press Cambridge, 2016.

[45] Y. Bengio *et al.*, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning*, 2009.

[46] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.

[47] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conference on Computer Vision*, pp. 499–515, Springer, 2016.

[48] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[49] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 539–546, 2005.

[50] C. L. Li, W. C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Advances in Neural Information Processing Systems*, pp. 2200–2210, 2017.

Weiwei LIN received a B.Eng. degree from Guangdong University of Technology in 2013 and an M. Sc. degree with distinction from the Hong Kong Polytechnic University in 2016. Since August 2016, he has been working toward a Ph.D. degree in electronic and information engineering at The Hong Kong Polytechnic University. His research interests include speaker recognition, transfer learning and deep learning.

Man-Wai MAK (M'93–SM'15) received a Ph.D. in electronic engineering from the University of Northumbria in 1993. He joined the Department of Electronic and Information Engineering at The Hong Kong Polytechnic University in 1993 and is currently an Associate Professor in the same department. He has authored more than 180 technical articles in speaker recognition, machine learning, and bioinformatics. Dr. Mak also coauthored a postgraduate textbook *Biometric Authentication: A Machine Learning Approach*, Prentice-Hall, 2005 and a research monograph *Machine Learning for Protein Subcellular Localization Prediction*, De Gruyter, 2015. He served as a member of the IEEE Machine Learning for Signal Processing Technical Committee in 2005–2007. He has served as an associate editor of IEEE/ACM Transactions on Audio, Speech and Language Processing. He is currently an associate editor of *Journal of Signal Processing Systems* and *IEEE Biometrics Compendium*. He also served as Technical Committee Members of a number of international conferences, including ICASSP and Interspeech, and gave a tutorial on machine learning for speaker recognition in Interspeech'2016. Dr. Mak's research interests include speaker recognition, machine learning, and bioinformatics.

Jen-Tzung CHIEN received his Ph.D. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, ROC, in 1997. He is now with the Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan, where he is currently a Chair Professor. He held the Visiting Professor position at the IBM T. J. Watson Research Center, Yorktown Heights, NY, in 2010. His research interests include machine learning, deep learning, speaker recognition, natural language processing and computer vision.

Dr. Chien served as the associate editor of the IEEE Signal Processing Letters in 2008-2011 and the tutorial speaker of the ICASSP in 2012, 2015, 2017, the INTERSPEECH in 2013, 2016, the COLING in 2018 and the general co-chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2017. He received the Best Paper Award of IEEE Automatic Speech Recognition and Understanding Workshop in 2011 and the AAPM Farrington Daniels Award in 2018. He is currently serving as an elected member of the IEEE Machine Learning for Signal Processing Technical Committee.