Generalized domain adaptation framework for parametric back-end in speaker recognition

Qiongqiong Wang, Koji Okabe, Kong Aik Lee, *Senior Member, IEEE,* and Takafumi Koshinaka *Member, IEEE*

***Abstract*—State-of-the-art speaker recognition systems comprise a speaker embedding front-end followed by a probabilistic linear discriminant analysis (PLDA) back-end. The effectiveness of these components relies on the availability of a large amount of labeled training data. In practice, it is common for domains (e.g., language, channel, demographic) in which a system is deployed to differ from that in which a system has been trained. To close the resulting gap, domain adaptation is often essential for PLDA models. Among two of its variants are Heavy-tailed PLDA (HT-PLDA) and Gaussian PLDA (G-PLDA). Though the former better fits real feature spaces than does the latter, its popularity has been severely limited by its computational complexity and, especially, by the difficulty, it presents in domain adaptation, which results from its non-Gaussian property. Various domain adaptation methods have been proposed for G-PLDA. This paper proposes a generalized framework for domain adaptation that can be applied to both of the above variants of PLDA for speaker recognition. It not only includes several existing supervised and unsupervised domain adaptation methods but also makes possible more flexible usage of available data in different domains. In particular, we introduce here two new techniques: (1) correlation-alignment in the model level, and (2) covariance regularization. To the best of our knowledge, this is the first proposed application of such techniques for domain adaptation w.r.t. HT-PLDA. The efficacy of the proposed techniques has been experimentally validated on NIST 2016, 2018, and 2019 Speaker Recognition Evaluation (SRE'16, SRE'18 and SRE'19) datasets.**

***Index Terms*—Speaker recognition, voice biometrics, domain adaptation, correlation alignment, regularization**

I. Introduction

SPEAKER recognition is the biometric task of recognizing a person from his/her voice on the basis of a small amount of speech utterance from that person [1]. Speaker embeddings are fixed-length continuous-value vectors that provide succinct characterizations of speakers' voices rendered in speech utterances. Recent progress in speaker recognition has achieved successful application of deep neural networks to derive deep speaker embeddings from speech utterances [2]–[5]. In a way similar to classical i-vectors [6], deep speaker embeddings live in a relatively simple Euclidean space in which distance can be measured far more easily than with much more complex input patterns. Techniques such as within-class covariance normalization (WCCN) [7], linear discriminant analysis (LDA) [8], and probabilistic linear discriminant analysis (PLDA) [9]–[11] can also be applied.

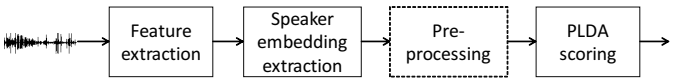

Fig. 1: State-of-the-art Speaker Recognition Pipeline

State-of-the-art text-independent speaker recognition systems composed of the speaker embedding front-end followed by a PLDA back-end, as illustrated in Fig. 1, have shown promising performance [12] [13]. The effectiveness of these components relies on the availability of a large amount of labeled training data, typically over one hundred hours of speech recordings consisting of multi-session recordings from several thousand speakers under differing conditions (e.g., w.r.t. recording devices, transmission channels, noise, reverberation, etc.). These knowledge sources contribute to the robustness of systems against such nuisance factors. The challenging problem of domain mismatch arises when a speaker recognition system is used in a different domain than that of its training data (e.g., different languages, demographics, etc.). It would be prohibitively expensive, however, to collect such a large amount of in-domain (InD) data for a new domain of interest for every application and then to retrain models. Most available resource-rich data that already exist will not match new domains of interest, i.e., most will be out-of-domain (OOD) data [14]–[16]. A more viable solution would be to adapt an already trained model using a smaller, and possibly unlabeled, set of in-domain data.

Domain adaptation techniques designed to adapt resource-rich OOD systems to produce good results in new domains have recently been studied with the aim of alleviating this problem [17]–[21]. Domain adaptation could be accomplished at different stages of the x-vector PLDA pipeline: 1) *data pooling*, 2) *speaker embedding compensation*, or 3) *PLDA parameter adaptation*. PLDA parameter adaptation is preferable in practice, as the same feature extraction and speaker embedding front-end can be used, while domain-adapted PLDA back-ends are used to cater to the condition in specific individual deployments [22]. It directly optimizes a model in an efficient way and does not require computationally expensive retraining with large-scale OOD data or the transformation of individual feature vectors.

There are two variants of PLDA: Gaussian PLDA (G-PLDA) and heavy-tailed PLDA (HT-PLDA). G-PLDA is the standard back-end for speaker recognizers that use speaker

Qiongqiong Wang and Kong Aik Lee are with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A^*STAR), Singapore. (e-mail: wang_qiongqiong@i2r.a-star.edu.sg, kongaik.lee@ieee.org).

Koji Okabe is with Biometrics Research Laboratories, NEC, Japan (e-mail: k-okabe@nec.com.

Takafumi Koshinaka is with the School of Data Science, Yokohama City University, Japan. He was with Biometrics Research Laboratories, NEC, Japan (e-mail: koshinak@yokohama-cu.ac.jp).

embeddings, assisted by a Gaussianization step involving length normalization (LN). HT-PLDA has been shown to be superior to G-PLDA in that it better fits real embedding spaces, though the computational cost is considerable [11]. It has subsequently been shown that embedding vectors could instead be Gaussianized via a simple LN procedure and provide performance comparable to that of an HT-PLDA with negligible extra computational cost [23]. Thus, G-PLDA with LN as the standard back-end for scoring text-independent speaker recognizers is the target for which model-level domain adaptation methods have been developed. At the same time, HT-PLDA, due to the computational complexity arising from its non-Gaussian property, has not achieved popularity, and it is especially difficult to adapt HT-PLDA's model parameters. It has, however, drawn researchers' attention because of a slightly simplified HT-PLDA with a computationally attractive alternative to that given by G-PLDA. This was achieved by a fast variational Bayes generative training algorithm and a fast scoring algorithm [24] [25]. The new type of HT-PLDA has demonstrated higher performance than G-PLDA + LN in NIST SRE'18 evaluations [22] [26] [27].

It is known that PLDA models are heavily affected by domain mismatch [22] [27] [28]. Many domain adaptation methods have been proposed to adapt G-PLDA models independently [19]–[21] [29] [30]. No previous work in domain adaptation, however, has given attention to HT-PLDA.

Our contributions in this paper are as follows: First, we present consistent formulations in both embedding-level and model-level domain adaptation. Existing works have presented domain adaptation only in either embedding-level or model-level domain adaptation. In this paper, we solve the data shift problem at the model level. Second, our comparison of embedding- and model-level domain adaptation reveals unique traits and shows that regularization is important to domain adaptation. Third, we propose a generalized form for semi-supervised domain adaptation that offers a way to use models or embeddings, and labeled or unlabeled data. We have also extended our previous work [29] - correlation-alignment-based interpolation and covariance regularization on the basis of linear interpolation [30] for robust domain adaptation. Finally, we demonstrate the above-mentioned three contributions in two back-ends and enable HT-PLDA to perform domain adaptation at the model level. We present an approximated formulation for unsupervised HT-PLDA adaptation based on state-of-the-art techniques for G-PLDA. We also review here a standard supervised adaptation method based on linear combination [30] and show how those adaptation techniques behave in popular settings [4] [5] [12]. To the best of our knowledge, this is the first such work on HT-PLDA adaptation.

The remainder of this paper is organized as follows. Section II reviews related work of domain adaptation. Section III reviews G-PLDA and HT-PLDA. Section IV introduces a proposed generalized framework for domain adaptation. Section V compares several unsupervised and supervised domain adaptation methods and shows their relationships to the generalized framework. Section VI describes our experimental setup, results, and analyses, and Section VII summarizes our work.

II. Related Work

Data pooling has been proposed, for example, to add a small amount of InD data (insufficient to train a model by itself) to a large amount of OOD data to train PLDA together [17]. Domain adaptation in this stage utilizes the speaker labels of the InD data. Thus, it is a kind of supervised domain adaptation. Both supervised and unsupervised domain adaptation methods can be applied to speaker embedding vectors. In adaptations at this level, statistical information about speaker embeddings in both the OOD and InD domains is used to process the OOD embeddings in order to compensate for shifts and rotations in embedding space caused by domain mismatch.

Both supervised and unsupervised domain adaptation methods can be applied at the speaker embedding level and model level. At the speaker embedding level, statistical information about speaker embeddings in both the OOD and InD domains is used to process the OOD embeddings in order to compensate for shifts and rotations in embedding space caused by domain mismatch. Inter dataset variability compensation (IDVC) has been proposed to compensate for dataset shift by constraining the shifts to a low dimensional subspace [18]. CORrelation ALignment (CORAL) adaption whitens and re-colors OOD embeddings using total covariance matrices calculated from OOD and InD embeddings [19]. Feature-Distribution Adaptor [21] and CORAL++ [31] further apply regularization on top of the second-order statistics alignment to avoid the influence of residual components and inaccurate information during adaptation. More recently, deep neural networks have been found to be effective in directly mapping speaker embeddings to another space [32]. At the model level, a PLDA model trained using OOD data can be adapted using either statistical information regarding the speaker embeddings in the InD data or parameters of another PLDA model trained using a limited amount of labeled InD data.

Among popular and effective ones are clustering methods [33], which label InD data using cluster labels, Kaldi domain adaptation [34] and CORAL+ [20], which utilize the total covariance matrix calculated from unlabeled InD data to update an OOD PLDA. Among these, supervised domain adaptation is more powerful than unsupervised. In [35], a maximum likelihood linear transformation has been proposed for transforming OOD PLDA parameters, so as to be closer to InD. A linear interpolation method has been proposed for combining parameters of PLDAs trained separately with OOD and InD data, so as to take advantage of both PLDAs [30]. In some applications, both labeled and unlabeled InD data exist. To make the best use of data and labels, we can apply semi-supervised domain adaptation methods. In [29], correlation-alignment-based interpolation has been proposed that utilizes the statistical information in the features of the InD data to update OOD PLDA before interpolation with an InD PLDA trained with labeled InD data.

III. Speaker verification in the embedding space

In state-of-the-art text-independent speaker recognition systems, the probabilistic linear discriminant analysis (PLDA) that was originally introduced in [9] [10] for face recognition

has become heavily employed for speaker embedding front-end [6] [11] [36] (see Fig. 1). PLDA decomposes the total variability into within-speaker and between-speaker variability. Some popular speaker embeddings are x-vectors [4] that utilize the time-delayed neural network and a statistic or attentive pooling layer to obtain fixed-length utterance-level representation. More sophisticated embeddings have also been proposed such as xi-vectors [37] which employ a posterior inference pooling to predict the frame-wise uncertainty of the input and propagate to the embedding vector estimation. There are two main variants of PLDAs: Gaussian PLDA (G-PLDA) and heavy-tailed PLDA (HT-PLDA).

A. Gaussian PLDA (G-PLDA)

G-PLDA can be seen as a special case of Joint Factor Analysis [38] with a single Gaussian component. Let $\boldsymbol{\phi}$ be a D-dimensional embedding vector (e.g., x-vector, xi-vector, etc.). We assume that vector $\boldsymbol{\phi}$ is generated from a linear Gaussian model [8], as follows [9] [10]:

$$P(\boldsymbol{\phi}|\mathbf{h},\mathbf{x}) = \mathcal{N}(\boldsymbol{\phi}|\boldsymbol{\mu}+\mathbf{F}\mathbf{h}+\mathbf{G}\mathbf{x},\boldsymbol{\Sigma}), \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ represents the global mean. The variables $\mathbf{h}$ and $\mathbf{x}$ are, respectively, the latent speaker and channel variables of d_h and d_x dimensions, while $\mathbf{F} \in \mathbb{R}^{D\times d_h}$ and $\mathbf{G} \in \mathbb{R}^{D\times d_x}$ are, respectively, the speaker and channel loading matrices, and the diagonal matrix $\boldsymbol{\Sigma}$ models residual variances. The between- and within-speaker covariance matrices $\{\boldsymbol{\Phi}_\mathrm{B}, \boldsymbol{\Phi}_\mathrm{W}\}$ can be derived from the latent speaker and channel variables as

$$\boldsymbol{\Phi}_\mathrm{B} = \mathbf{F}\mathbf{F}^\mathsf{T}, \boldsymbol{\Phi}_\mathrm{W} = \mathbf{G}\mathbf{G}^\mathsf{T} + \boldsymbol{\Sigma}. \tag{2}$$

The speaker conditional distributions

$$P(\boldsymbol{\phi}|\mathbf{h}) = \mathcal{N}(\boldsymbol{\phi}|\boldsymbol{\mu}+\mathbf{F}\mathbf{h}, \boldsymbol{\Phi}_\mathrm{w}) \tag{3}$$

have a common within-speaker covariance matrix $\boldsymbol{\Phi}_\mathrm{W}$. The speaker variables have a Gaussian prior:

$$P(\mathbf{h}) = \mathcal{N}(\mathbf{h}|\mathbf{0},\mathbf{I}). \tag{4}$$

B. Heavy-tailed PLDA (HT-PLDA)

Unlike G-PLDA, which assumes Gaussian priors of both channel and speaker factors $\mathbf{x}$ and $\mathbf{h}$ in (1), the heavy-tailed version of PLDA (HT-PLDA) replaces Gaussian distributions with Student's t-distributions. In [24] [25], a fast training and scoring algorithm for a simplified HT-PLDA back-end was proposed, with speed comparable to that with G-PLDA. For every speaker, a hidden speaker identity variable, $\mathbf{h} \in \mathbb{R}^{d_h}$, is drawn from the standard normal distribution. The heavy-tailed behavior is obtained by having a precision scaling factor, $\lambda > 0$, drawn from a gamma distribution, $\mathcal{G}(\alpha,\beta)$ parametrized by $\alpha = \beta = \frac{\nu}{2} > 0$. The parameter ν is known as the *degrees of freedom* [8] [11]. Given the hidden variables, the embedding vectors $\boldsymbol{\phi}$ are described by a multivariate normal:

$$P(\boldsymbol{\phi}|\mathbf{h},\lambda) = \mathcal{N}(\boldsymbol{\phi}|\mathbf{F}\mathbf{h}, (\lambda\mathbf{W})^{-1}) \tag{5}$$

where $\mathbf{F} \in \mathbb{R}^{D\times d_h}$ is the speech loading matrix, $\mathbf{W} \in \mathbb{R}^{D\times D}$ is a positive definite *precision matrix*, and D is the speaker embedding dimension. The model parameters are $\nu, \mathbf{F}, \mathbf{W}$. This model is a simplification of the HT-PLDA model in [11]. Because of the heavy-tailed speaker identity variables, (5) can be represented in a multivariate t-distribution [8] [25]:

$$P(\boldsymbol{\phi}|\mathbf{h}) = \mathcal{T}(\boldsymbol{\phi}|\mathbf{F}\mathbf{h}, \mathbf{W}, \nu). \tag{6}$$

In the condition in which $D > d$, and $\mathbf{F}^\mathsf{T}\mathbf{W}\mathbf{F}$ is invertible, (6) can be proved to be proportional to another t-distribution, with increased degrees of freedom $\nu' = \nu + D - d$:

$$P(\boldsymbol{\phi}|\mathbf{h}) = \mathcal{T}(\mathbf{h}|\hat{\mathbf{h}}, \mathbf{B}, \nu') \tag{7}$$

where $\hat{\mathbf{h}}$ and $\mathbf{B}$ are utterance dependent:

$$\hat{\mathbf{h}} = \mathbf{B}^{-1}\mathbf{a}, \qquad \mathbf{B} = b\mathbf{B}_0,$$
$$\mathbf{a} = b\mathbf{F}^\mathsf{T}\mathbf{W}\phi, \qquad b = \frac{\nu + D - d}{\nu + \boldsymbol{\phi}^\mathsf{T}\mathbf{G}\boldsymbol{\phi}},$$

while $\mathbf{B}_0$ and $\mathbf{G}$ are utterance-independent parameters that only need to be calculated once and are constant for all the speaker embeddings:

$$\mathbf{B}_0 = \mathbf{F}^\mathsf{T}\mathbf{W}\mathbf{F}, \qquad \mathbf{G} = \mathbf{W} - \mathbf{W}\mathbf{F}\mathbf{B}_0^{-1}\mathbf{F}^\mathsf{T}\mathbf{W}. \tag{8}$$

In this paper, we propose domain adaptation for HT-PLDA based on this algorithm.

C. Domain adaptation based on speaker embeddings

In speaker recognition, when test data is mismatched with the domain in which the model has been trained, performance degrades drastically. In practice, collecting a large amount of data with speaker labels in the target domain is expensive. An alternative solution is to do domain adaptation from a source domain to a target domain in which labeled training data is scarce. Two typical domain adaptation scenarios are: (1) supervised domain adaptation using a small amount of in-domain (InD) data with speaker labels, and (2) unsupervised domain adaptation using relatively larger amount of unlabeled InD data.

The final goal of domain adaptation based on speaker embeddings is to perform covariance matching from a source domain to a target domain. In supervised domain adaptation, a new set of between-speaker and within-speaker covariance matrices for the target domain can be calculated from a limited amount of labeled data, though these matrices may not be very accurate or reliable. They are used to adapt the out-of-domain (OOD) covariance matrices [30]. In unsupervised domain adaptation, only a total covariance matrix (i.e., the summation of the between-speaker and within-speaker covariance matrices) can be calculated from the unlabeled data, and not the two covariance matrices separately. The total covariance matrix is often used for adaptation [20] [21] [34] [39].

Domain adaptation can be applied via data, embedding compensation, and/or model adaptation. At the data level, InD data with speaker labels is added to a pool of OOD data to train a PLDA model [17], which is only feasible as a supervised method. At the embedding level, domain adaptation is used to rotate and re-scale the embeddings in the source domain

by multiplying it with a transformation matrix $\mathbf{A}$ to fit a distribution better in the target domain:

$$\boldsymbol{\phi}^* = \mathbf{A}\boldsymbol{\phi}. \tag{9}$$

At the model level, between-speaker and within-speaker covariance matrices can be transformed or fused. This is popular in practice since there is no need for extra memory to store the large amount of the source domain training data other than that for the trained OOD model. In principle, embedding compensation here can be considered equivalent to model-level domain adaptation, as the transformation in embedding in (9) results in a change in total covariance matrices:

$$\boldsymbol{\Phi}^* = \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^\mathsf{T}. \tag{10}$$

The equivalency stands in the condition in which the embeddings are used directly to train a back-end model without any pre-processing. In this paper, we present speaker embedding compensation methods in the form of model adaptation and compare them with other model level methods, utilizing the relationships shown in (9) and (10).

In both supervised and unsupervised domain adaptation, the goal is to propagate the variance, i.e., uncertainty seen in the InD data to the adapted model. This means that regularization [20] between covariance matrices will be important to guarantee the uncertainty increase. In addition, it can also improve the robustness of some fusion-based domain adaption models against interpolation weights. In some applications, both labeled and unlabeled InD data exist. To make the most use of the data and the labels, we can apply semi-supervised domain adaptation methods. In [29], correlation-alignment-based interpolation was proposed that utilized the statistical information in the features of InD data to update OOD PLDA before interpolation with an InD PLDA trained with labeled InD data.

IV. Generalized Framework for PLDA adaptation

We propose a generalized framework for robust PLDA domain adaptation in both unsupervised and supervised manners:

$$\boldsymbol{\Phi}^+ = \alpha\boldsymbol{\Phi}_0 + \beta\Gamma_{\max}(\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2) \tag{11}$$

where $\boldsymbol{\Phi}^+$ represents the between and within-speaker covariance matrix of the domain-adapted PLDA. $\boldsymbol{\Phi}_0$, $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are covariance matrices from up to three PLDA models, and $\{\alpha, \beta\}$ are the weighting parameters. The regularization function

$$\Gamma_{\max}(\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2) = \mathbf{B}^\mathsf{T} \max(\mathbf{E}, \mathbf{I})\mathbf{B} \tag{12}$$

can be considered a processed covariance matrix computed taking the information of both covariance matrices and that guarantees the variability to be larger than both $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$. The operator $\max(\cdot)$ takes the larger values from two matrices in an element-wise manner. The diagonal matrix $\mathbf{E}$ and the identical matrix $\mathbf{I}$ correspond to $\{\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2}\}$, respectively, after the projection of $\mathbf{B}^\mathsf{T}(\cdot)\mathbf{B}$:

$$\mathbf{B}^\mathsf{T}\boldsymbol{\Phi}_1\mathbf{B} = \mathbf{E} \qquad \mathbf{B}^\mathsf{T}\boldsymbol{\Phi}_2\mathbf{B} = \mathbf{I}. \tag{13}$$

In the case where the between- and within-speaker covariance matrices are real-value symmetric matrices with full ranks, they can be eigen-decomposed (EVD) with d number of real non-zero eigenvalues and corresponding eigenvectors. Thus, $\{\mathbf{B}, \mathbf{E}\}$ are obtained by a simultaneous diagonalization of $\{\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2}\}$ [40]:

$$\begin{aligned} \{\mathbf{Q}, \boldsymbol{\Lambda}\} &\leftarrow \mathrm{EVD}(\boldsymbol{\Phi}_1) \\ \{\mathbf{P}, \mathbf{E}\} &\leftarrow \mathrm{EVD}(\boldsymbol{\Lambda}^{-1/2}\mathbf{Q}^\mathsf{T}\boldsymbol{\Phi}_2\mathbf{Q}\Lambda^{-1/2}) \\ \mathbf{B} &= \mathbf{Q}\boldsymbol{\Lambda}^{-1/2}\mathbf{P} \end{aligned}$$

$\mathbf{B}$ is shown to be a square matrix with real-value elements and $\mathbf{B}\mathbf{B}^\mathsf{T} = \mathbf{I}$, and thus, is an orthogonal matrix.

The generalized framework (11) is a linear interpolation of $\boldsymbol{\Phi}_0$ and a regularization result from $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$. To have a better interpretation, we define $\boldsymbol{\Phi}_0$ as a covariance matrix of a base PLDA from which a new PLDA has been adapted. Though $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$, theoretically, can be swapped, we interpret them differently. When a new PLDA (corresponding to $\boldsymbol{\Phi}_1$ here) is available to develop the base PLDA ($\boldsymbol{\Phi}_0$), instead of a direct linear interpolation between them, we regularize $\boldsymbol{\Phi}_1$ using the covariance $\boldsymbol{\Phi}_2$ of another PLDA as the reference. Thus, PLDAs that correspond to $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are defined as a developer PLDA and a reference PLDA, respectively. This generalized framework enables us to combine several existing supervised and unsupervised domain adaptations into a single formulation, which we show in Section V that follows.

For G-PLDA where the dimension is set to the same as that of the embedding vectors, with the number of speakers in the training data larger than the dimension, the between- and within-speaker covariance matrices are real-value symmetric matrices with full ranks. Therefore, regularization (12) can be applied as mentioned above. Next, let us give an estimation of domain adaptation for HT-PLDA. In the new algorithm of HT-PLDA introduced in Section III-B, the approximate between- and within-speaker covariance matrices are

$$\boldsymbol{\Phi}_\mathbf{B} = \mathbf{F}\mathbf{F}^\mathsf{T}, \qquad \boldsymbol{\Phi}_\mathbf{W} = (\lambda\mathbf{W})^{-1}. \tag{14}$$

It should be noted that the within-speaker covariance matrix is a variable dependent on individual utterances because λ is the hidden precision scaling factor that varies utterance by utterance. We assume that the scalar parameter ν and matrix $\mathbf{W^{-1}}$ are independent of each other, and we approximate the domain adaptation of the within-speaker covariance matrix as the adaptation of the matrix $\mathbf{W^{-1}}$ and the scalar parameter ν separately. We propose to adapt $\mathbf{W^{-1}}$ in the same way as with the between- and within-speaker covariance matrices in (11). In this paper, we do not discuss the adaptation of the parameter ν, as [24] [25] have shown it to be a pre-defined parameter.

Unlike the within-speaker covariance matrix, the between-speaker covariance in (14) is a fixed matrix once the HT-PLDA model has been trained. Because the new algorithm for HT-PLDA [24] [25] is based on the assumption of $d_h \ll D$, the between-speaker covariance matrix is rank-deficient. This causes the inapplicability of the proposed covariance regularization technique in (12). Studies have shown that between-speaker covariance is less affected by domain mismatch than is within-speaker covariance. Thus, in this paper for HT-PLDA, we investigate the effect of the proposed covariance

TABLE I: The special cases derived from the general form.

	Method	Φ_0	Φ_1	Φ_2	Eq.
1	LIP [30]	$\mathbf{\Phi}_{\mathrm{I}}$	$\mathbf{\Phi}_{\mathrm{O}}$	$\mathbf{\Phi}_{\mathrm{O}}$	(16)
2	CORAL [19]	0	$\mathbf{\Phi}_{\mathrm{CORAL}}$	$\mathbf{\Phi}_{\mathrm{CORAL}}$	(24)
3	CIP [29]	$\mathbf{\Phi}_{\mathrm{I}}$	$\mathbf{\Phi}_{\mathrm{CORAL}}$	$\mathbf{\Phi}_{\mathrm{CORAL}}$	(17)(25)
4	CORAL+ [20]	$\mathbf{\Phi}_{\mathrm{O}}$	$\mathbf{\Phi}_{\mathrm{CORAL}}$	$\mathbf{\Phi}_{\mathrm{O}}$	(29)(25)
5	Kaldi [34]	$\mathbf{\Phi}_{\mathrm{O}}$	$\mathbf{C}_{\mathrm{I}}$	$\mathbf{\Phi}_{\mathrm{O}}^{b}+\mathbf{\Phi}_{\mathrm{O}}^{w}$	(30)
6	LIP reg [29]	$\mathbf{\Phi}_{\mathrm{I}}$	$\mathbf{\Phi}_{\mathrm{O}}$	$\mathbf{\Phi}_{\mathrm{I}}$	(34)
7	CIP reg [29]	$\mathbf{\Phi}_{\mathrm{I}}$	$\mathbf{\Phi}_{\mathrm{CORAL}}$	$\mathbf{\Phi}_{\mathrm{I}}$	(35)(25)
8	FDA [21]	0	$\mathbf{\Phi}_{\mathrm{I,pseudo,1}}$	$\mathbf{\Phi}_{\mathrm{I,pseudo,1}}$	(36)
9	Kaldi* [21]	0	$\mathbf{\Phi}_{\mathrm{I,pseudo,2}}$	$\mathbf{\Phi}_{\mathrm{I,pseudo,2}}$	(25)(37)

regularization technique only in the within-speaker covariance matrix.

V. Relationships and extensions to existing methods

The generalized framework can be summarized in terms of the next three main factors: i) interpolation of covariance matrices, ii) correlation alignment, and iii) covariance regularization. In this section, we introduce several existing special cases of the general framework, as listed in Tab. I, and show their relationships with each other from the perspective of the three factors. The relationship diagram is briefly shown in Fig. 2, and their corresponding equations mentioned in this Section are listed in Tab. II) for fast reference.

A. Interpolation of covariance matrices

In a special case in which $\mathbf{\Phi}_1 = \mathbf{\Phi}_2$ in (11), meaning that they are from the same PLDA model, the regularization term $\Gamma_{max}(\cdot)$ is equivalent to the covariance matrix itself. Thus, the generalized framework presents a simple interpolation with a weight α of PLDA parameters, i.e., between- and within-speaker covariance of two PLDAs

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + \beta\mathbf{\Phi}_1. \tag{15}$$

In common domain adaptation scenarios, a sufficient amount of labeled OOD data are always available. InD data with labels is often insufficient, while unlabeled InD data is easier to access and obtain. In the first scenario, in which the InD data is labeled but insufficient, an InD PLDA model can be trained but will be rather unreliable or not be expected to perform well. It can, however, be interpolated with an OOD PLDA model

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{O}} \tag{16}$$

and play an important role leading the OOD model in the direction of the target domain. It has been proposed as a supervised method in [30] and is shown as special case 1 linear interpolation (LIP) in Tab. I.

For the same scenario, the interpolation can be also applied to the InD model with a pseudo-InD model

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{I,pseudo}}, \tag{17}$$

where $\mathbf{\Phi}_{\mathrm{I,pseudo}}$ represents the preliminarily adapted PLDA model. It is supposed to be closer to a true InD PLDA and fit the target domain better than an OOD PLDA. Therefore, replacing the OOD model in (16) with the pseudo-Ind model is expected to result in a more reliable PLDA. More about the pseudo-InD model will be introduced later in this section.

In the second scenario, in which the InD data is unlabeled, an interpolation still can be conducted between an OOD PLDA and a pseudo-IND PLDA

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{O}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{I,pseudo}}. \tag{18}$$

The pseudo-InD PLDA here is a preliminarily adapted model from an OOD PLDA model using InD data in an unsupervised manner.

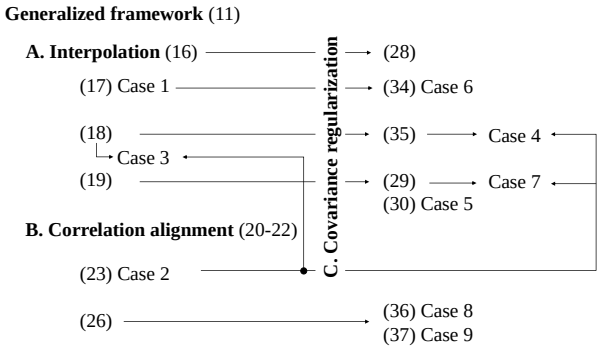

Fig. 2: The relationships between the special cases from the perspective of the three main factors of the generalized framework. (*) shows the equation index.

TABLE II: Relevant equation for fast reference for the relationship diagram in Fig. 2.

	Eq.	Equation
Generalized framework	11	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + \beta\Gamma_{\max}(\mathbf{\Phi}_1, \mathbf{\Phi}_2,)$
A. Interpolation	16	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{O}}$
	17	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{I,pseudo}}$
	18	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{O}} + (\mathbf{1}-\alpha)\mathbf{\Phi}_{\mathrm{I,pseudo}}$
	19	$\boldsymbol{\phi}_{\mathrm{I,pseudo}} = \mathbf{A}\boldsymbol{\phi}_{\mathrm{O}}$
B. Correlation alignment	20	$\mathbf{C}_{\mathrm{I,pseudo}} = \mathbf{A}\mathbf{C}_{\mathrm{O}}\mathbf{A}^{\mathsf{T}} = \mathbf{C}_{\mathrm{I}}$
	21	$\mathbf{\Phi}_{\mathrm{I,pseudo}} = \mathbf{A}\mathbf{\Phi}_{\mathrm{O}}\mathbf{A}^{\mathsf{T}}$
	22	$\mathbf{A} = \mathrm{C}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}$
	23	$\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}} = \mathbf{Q}_{\mathrm{O}}\mathbf{\Lambda}_{\mathrm{O}}^{-\frac{1}{2}}\mathbf{Q}_{\mathrm{O}}{}^{\mathsf{T}}, \mathbf{C}_{\mathrm{I}}^{\frac{1}{2}} = \mathbf{Q}_{\mathrm{I}}\mathbf{\Lambda}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{Q}_{\mathrm{I}}{}^{\mathsf{T}}$
	26	$\boldsymbol{\phi}_{\mathrm{I,pseudo}} = \mathbf{C}_{\mathrm{O}}^{\frac{1}{2}}\mathbf{P}\Delta^{\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\boldsymbol{\phi}_{\mathrm{O}}$
C. Covariance regularization	28	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + (1-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_1, \mathbf{\Phi}_0)$
	29	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{O}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{I,pseudo}}, \mathbf{\Phi}_{\mathrm{O}})$
	30	$\mathbf{\Phi}_{\mathrm{tot}}^{+} = \alpha\mathbf{\Phi}_{\mathrm{tot,O}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{C}_{\mathrm{I}}, \mathbf{\Phi}_{\mathrm{tot,O}})$
	34	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{O}}, \mathbf{\Phi}_{\mathrm{I}})$
	35	$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{I,pseudo}}, \mathbf{\Phi}_{\mathrm{I}})$
	36	$\boldsymbol{\phi}_{\mathrm{I,pseudo}} = \mathbf{C}_{\mathrm{O}}^{\frac{1}{2}}\mathbf{P}\hat{\Delta}^{\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\boldsymbol{\phi}_{\mathrm{O}}$
	37	$\boldsymbol{\phi}_{\mathrm{I,pseudo}} = \mathbf{\Phi}_{\mathrm{tot,O}}^{\frac{1}{2}}\mathbf{P}\hat{\Delta}^{\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}\boldsymbol{\phi}_{\mathrm{O}}$ $\mathbf{P}\hat{\Delta}\mathbf{P}^{\mathsf{T}} = \Gamma_{max}(\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}\mathbf{C}_{\mathrm{I}}\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}, \mathbf{I})$

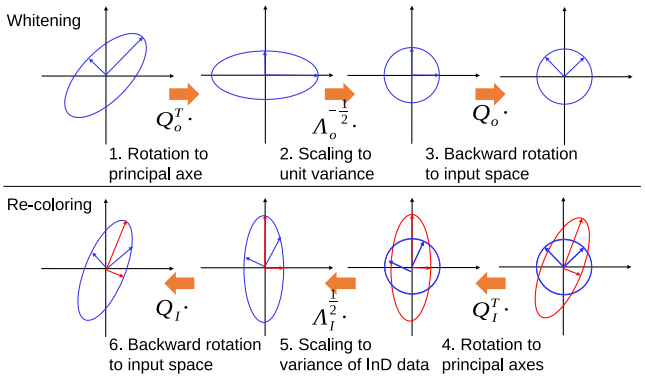

Fig. 3: Illustration of whitening and re-coloring with ZCA transformations in CORAL

B. Correlation alignment

Covariance alignment aims to align second-order statistics, i.e., covariance matrices, of OOD embeddings so that they will match InD embeddings while maintaining the good properties that OOD PLDA learned from a large amount of data. In formulaic terms, it would be to find a matrix $\mathbf{A}$ so that the pseudo-InD embeddings $\phi_{\mathrm{I,pseudo}}$ that are transformed from the OOD embeddings

$$\phi_{\mathrm{I,pseudo}} = \mathbf{A}\phi_{\mathrm{O}} \tag{19}$$

will have the total covariance matrix $\mathbf{A}\mathbf{C}_{\mathrm{O}}\mathbf{A}^{\mathsf{T}}$ close to or almost equivalent to that of InD. To calculate the transformation matrix $\mathbf{A}$, we set

$$\mathbf{C}_{\mathrm{I,pseudo}} \leftarrow \mathbf{A}\mathbf{C}_{\mathrm{O}}\mathbf{A}^{\mathsf{T}} = \mathbf{C}_{\mathrm{I}}, \tag{20}$$

where $\mathbf{C}_{\mathrm{O}}$ and $\mathbf{C}_{\mathrm{I}}$ are the empirical total covariance matrices estimated from the OOD dataset X_{O} and InD data sets X_{I}. It is commonly known that a linear transformation on a normally distributed vector leads to an equivalent transformation on the mean vector and covariance matrix of its density function. Trained with the pseudo-InD embeddings obtained in (19), the between- and within-speaker covariance matrices of the pseudo-InD PLDA have the corresponding transformation

$$\mathbf{\Phi}_{\mathrm{I,pseudo}} = \mathbf{A}\mathbf{\Phi}_{\mathrm{O}}\mathbf{A}^{\mathsf{T}}. \tag{21}$$

Therefore, in this paper, we consider any embedding-level adaptation using a transformation to be equivalent to model-level adaptation.

To satisfy (20), one ample algorithm is

$$\mathbf{A} = \mathrm{C}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}, \tag{22}$$

which consists of whitening $\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}$ followed by re-coloring $\mathbf{C}_{\mathrm{I}}^{\frac{1}{2}}$:

$$\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}} = \mathbf{Q}_{\mathrm{O}}\mathbf{\Lambda}_{\mathrm{O}}^{-\frac{1}{2}}\mathbf{Q}_{\mathrm{O}}{}^{\mathsf{T}}, \qquad \mathbf{C}_{\mathrm{I}}^{\frac{1}{2}} = \mathbf{Q}_{\mathrm{I}}\mathbf{\Lambda}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{Q}_{\mathrm{I}}{}^{\mathsf{T}}. \tag{23}$$

Here, $\mathbf{Q}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues pertaining to the covariance matrices.

Thus the whitening and re-coloring processes can be interpreted as the zero-phase component analysis (ZCA) transformation [41] with six steps, as shown in Fig. 3. For the whitening process, the OOD embeddings ϕ_{O} are first rotated to the principal axes of $\mathbf{C}_{\mathrm{O}}$ space, then scaled into a unit variance, and finally rotated back to the input space. For the re-coloring process, similarly, the normalized embeddings are first rotated to the principal axes of $\mathbf{C}_{\mathrm{I}}$ space, then re-scaled to have the variance of InD data, and, finally, rotated backward to the input space. As opposed to principal component analysis (PCA) and Cholesky whitening (and re-coloring), ZCA preserves the maximal similarity of the transformed feature to the original space. Since no speaker label is used, the embedding-level domain adaptation

$$\phi_{\mathrm{I,pseudo}} = \mathrm{C}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\phi_{\mathrm{O}} \tag{24}$$

using (22) in (19) is an unsupervised embedding-level domain adaptation, and it is referred to as correlation alignment (CORAL) [39]. It is shown as special case 2 in Tab. I. Thus, when the PLDA trained using CORAL adapted embeddings as a pseudo-InD model

$$\mathbf{\Phi}_{\mathrm{I,pseudo}} = \mathrm{C}_{\mathrm{I}}^{\frac{1}{2}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\mathbf{\Phi}_{\mathrm{O}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\mathrm{C}_{\mathrm{I}}^{\frac{1}{2}} \tag{25}$$

for the linear interpolation with an InD model in (17), the process is referred to as correlation-alignment-based interpolation (CIP), as is proposed in [29]. It is shown as special case 3 in Tab. I.

Another embedding transformation which still satisfies the correlation alignment in (20) is

$$\phi_{\mathrm{I,pseudo}} = \mathbf{C}_{\mathrm{O}}^{\frac{1}{2}}\mathbf{P}\Delta^{\frac{1}{2}}\mathbf{P}^{\mathrm{T}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\phi_{\mathrm{O}}, \tag{26}$$

where $\mathbf{P}$ and Δ are from an eigenvalue decomposition

$$\mathbf{P}\Delta\mathbf{P}^{\mathrm{T}} = \mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\mathbf{C}_{\mathrm{I}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}. \tag{27}$$

In the transformation, it also uses the empirical total covariance matrices from the OOD data and InD data. It first whitens the features with $\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}$, just as that in CORAL, and then recolors them with $\mathbf{C}_{\mathrm{O}}^{\frac{1}{2}}[\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\mathbf{C}_{\mathrm{I}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}]^{\frac{1}{2}}$. This is the fundamental idea of the feature-distribution adaptor (FDA) proposed in [21].

C. Covariance regularization

The central idea in domain adaptation is to propagate the uncertainty seen in the data to the model. Neither the adaptation equations for linear interpolation in Section V-A nor those for correlation alignment in Section V-B guarantee that the variance, and therefore the uncertainty, will increase. The domain adaptations for PLDA work with multiple covariance matrices, including the total covariance calculated from the available data sets and the between- and within-speaker covariance of the trainable models. Thus, regularization is introduced to propagate the uncertainty seen in any two PLDA models. As shown in Section IV, $\Gamma_{max}(\mathbf{\Phi}_1, \mathbf{\Phi}_2)$ is a regularization function that guarantees the variability to be larger than both $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$. This insures the lower boundary of the performance of domain adaptation methods.

Algorithm 1: The Kaldi algorithm [34] for unsupervised adaptation of PLDA.

Input Out-of-domain PLDA matrices $\{\mathbf{\Phi}_{\mathrm{B,O}}, \mathbf{\Phi}_{\mathrm{W,O}}\}$
In-domain data X_I
Adaptation hyper-parameters $\{\gamma, \beta\}$
Output Adapted covariance matrices $\{\mathbf{\Phi}_{\mathrm{B}}, \mathbf{\Phi}_{\mathrm{W}}\}$
Estimate empirical covariance matrix from the in-domain data X_I
$\mathbf{C}_{\mathrm{I}} = \mathrm{Cov}(X_I)$
Compute out-of-domain covariance matrix
$\mathbf{\Phi}_{\mathrm{tot,O}} = \mathbf{\Phi}_{\mathrm{B,O}} + \mathbf{\Phi}_{\mathrm{W,O}}$
Find $\{\mathbf{B}, \mathbf{E}\}$ via simultaneous diagonalization of $\mathbf{\Phi}_{\mathrm{tot,O}}$ and $\mathbf{C}_{\mathrm{I}}$
$\{\mathbf{Q}, \mathbf{\Lambda}\} \leftarrow \mathbf{EVD}(\mathbf{C}_{\mathrm{O}})$
$\{\mathbf{P}, \mathbf{E}\} \leftarrow \mathbf{EVD}(\mathbf{\Lambda}^{-1/2}\mathbf{Q}^{\mathsf{T}}\mathbf{C}_{\mathrm{I}}\mathbf{Q}\mathbf{\Lambda}^{-1/2})$
$\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}^{-1/2}\mathbf{P}$
Kaldi unsupervised adaptation of PLDA, $\gamma + \beta <= 1$
$\mathbf{\Phi}_{\mathrm{W}}^{+} = \mathbf{\Phi}_{\mathrm{W,O}} + \beta_{\mathrm{W}}\mathbf{B}^{-\mathsf{T}} \max(\mathbf{E} - \mathbf{I})\mathbf{B}^{-1}$
$\mathbf{\Phi}_{\mathrm{B}}^{+} = \mathbf{\Phi}_{\mathrm{B,O}} + \beta_{\mathrm{B}}\mathbf{B}^{-\mathsf{T}} \max(\mathbf{E} - \mathbf{I})\mathbf{B}^{-1}$.
Notation $\mathrm{EVD}(\cdot)$ returns a matrix of eigenvectors and the corresponding eigenvalues in a diagonal matrix.

1) Covariance regularization for linear interpolations: For linear interpolation-based methods, performance is strongly affected by interpolation weights, which should be correlated to the performance of the individual model. In practice, however, tuning weights is often unfeasible. Applying covariance regularization to the linear interpolation in (15)

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + (1-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_1, \mathbf{\Phi}_0) \tag{28}$$

guarantees the adapted model to have a variance larger than the reference model $\mathbf{\Phi}_0$. In the case in which one covariance is larger than the other in their common space for all dimensions, the two extreme cases for (28) are 1) a simple interpolation $\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + (1-\alpha)\mathbf{\Phi}_1$, and 2) no domain adaptation: $\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_0 + (1-\alpha)\mathbf{\Phi}_0 = \mathbf{\Phi}_0$.

Regularization is expected to take a more important role when the interpolation is done in an unsupervised manner, as, for example, between an OOD PLDA model and an unsupervised pseudo-InD PLDA in (18)

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{O}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{I,pseudo}}, \mathbf{\Phi}_{\mathrm{O}}). \tag{29}$$

When CORAL [39] is used to train the pseudo-InD PLDA as shown in (25) for the pseudo-InD between- and within-speaker covariance matrices $\mathbf{\Phi}_{\mathrm{I,pseudo}}$, (29) is referred as CORAL+ proposed in [20]. It is shown as special case 4 in Tab. I. The procedure is illustrated in Algorithm 2 in Appendix A in [42].

Rather than predicting the between- and within-speaker covariance directly but independently, we can alternatively first predict the total covariance Φ_{tot} of the adapted model

$$\mathbf{\Phi}_{\mathrm{tot}}^{+} = \alpha\mathbf{\Phi}_{\mathrm{tot,O}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\mathbf{max}}(\mathbf{C}_{\mathrm{I}}, \mathbf{\Phi}_{\mathrm{tot,O}}) \tag{30}$$

and then distribute it to the between- and within-speaker covariance matrices. Here, regularization is applied to the InD empirical total covariance matrix $\mathbf{C}_{\mathrm{I}}$ calculated from the unlabeled InD data and the total covariance of the OOD PLDA model by summing up the between- and within-speaker covariance

$$\mathbf{\Phi}_{\mathrm{tot,O}} = \mathbf{\Phi}_{\mathrm{B,O}} + \mathbf{\Phi}_{\mathrm{W,O}}. \tag{31}$$

With regularization, the model guarantees that the total variances of $\mathbf{\Phi}_{\mathrm{tot}}$ will increase over that of $\mathbf{\Phi}_{\mathrm{tot,O}}$ before the adaptation. After the adapted total covariance is obtained, we split it into the adapted between- and within-speaker covariance matrices:

$$\mathbf{\Phi}_{\mathrm{W}}^{+} = \alpha\mathbf{\Phi}_{\mathrm{W,O}} + \beta_{\mathrm{W}}\mathbf{\Gamma}_{\mathbf{max}}(\mathbf{C}_{\mathrm{I}}, \mathbf{\Phi}_{\mathrm{tot,O}}) \tag{32}$$

$$\mathbf{\Phi}_{\mathrm{B}}^{+} = \alpha\mathbf{\Phi}_{\mathrm{B,O}} + \beta_{\mathrm{B}}\mathbf{\Gamma}_{\mathbf{max}}(\mathbf{C}_{\mathrm{I}}, \mathbf{\Phi}_{\mathrm{tot,O}}). \tag{33}$$

Here, $\beta_{\mathrm{B}} + \beta_{\mathrm{W}} = (1-\alpha)$. This adaptation was implemented as an unsupervised domain adaptation for Gaussian PLDA (G-PLDA) in the Kaldi toolbox [34]. It is shown as special case 5 in Tab. I. The procedure is illustrated in Algorithm 1.

Both Kaldi's domain adaptation and CORAL+ are unsupervised domain adaptation methods that adapt OOD between- and within-speaker covariance matrices using unlabeled InD data. They utilize regularization to guarantee variance increases by choosing the larger value between two covariance matrices that represent the source domain, i.e., OOD, and the target domain, i.e., InD, respectively, in their simultaneous diagonalized subspace. They both use the empirical InD total covariance calculated from the InD data. The differences are: CORAL+ aims at variance increases in each of the between- and within-speaker covariances, while Kaldi's aims at the variance increase only in the total covariance. There is no guarantee of variance increases in both between- and within-speaker covariance. In regularization, Kaldi's uses empirical InD total covariance $\mathbf{C}_{\mathrm{I}}$ directly, while CORAL+ uses pseudo InD within- and between-speaker covariance, i.e., uses the empirical InD total covariance indirectly. In Kaldi's method, after the total covariance has been adapted, two hyper-parameters $\{\beta_{\mathrm{B}}, \beta_{\mathrm{W}}\}$ are set to determine the portions of the variance increases in the total covariance for the between- and within-speaker covariance. The two hyper-parameters are restricted by each other and follow $\beta_{\mathrm{B}} + \beta_{\mathrm{W}} \leq 1$. CORAL+'s two hyper-parameters are independent from each other.

Regularization can also be applied to the supervised linear interpolation scenario. Here, we extend the linear interpolations in (16) and (17) to

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{O}}, \mathbf{\Phi}_{\mathrm{I}}), \tag{34}$$

which is referred to as LIP with regularization (LIP reg) as proposed in [29] (special case 6 in Tab. I), and

$$\mathbf{\Phi}^{+} = \alpha\mathbf{\Phi}_{\mathrm{I}} + (\mathbf{1}-\alpha)\mathbf{\Gamma}_{\max}(\mathbf{\Phi}_{\mathrm{I,pseudo}}, \mathbf{\Phi}_{\mathrm{I}}), \tag{35}$$

respectively. These two cases set the InD PLDA as the reference model for comparison purposes in order to confirm that applying regularization in the interpolation will ensure the uncertainty increase. When we again use CORAL [39] to train the pseudo-InD PLDA as shown in (25), (35) becomes the CIP with regularization (CIP reg) proposed in [29]. It is shown as special case 7 in Tab. I.

2) Reintolarization in covariance alignment: The regularization can also be applied to covariance alignment since it works with multiple covariance matrices as well. Rather than looking for an embedding-level transformation for the covariance alignment, as shown in (20) in Section V-B, it seeks for a transformation so that the adapted OOD embeddings will have the covariance with a guaranteed uncertainty increase. We can modify the embedding transformation in (26) to

$$\phi_{\mathrm{I,pseudo}} = \mathbf{C}_{\mathrm{O}}^{\frac{1}{2}}\mathbf{P}\hat{\Delta}^{\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{C}_{\mathrm{O}}^{-\frac{1}{2}}\phi_{\mathrm{O}} \tag{36}$$

by replacing the original Δ with a regularization $\hat{\Delta} = max(\mathbf{I}, \Delta)$. Its two extremes are $\hat{\Delta} = \mathbf{I}$, which corresponds to $\phi_{\mathrm{I,pseudo}} = \phi_{\mathrm{O}}$, meaning no adaptation is applied, and $\hat{\Delta} = \Delta$, where the embedding transformation is equivalent to (26). The algorithm is an unsupervised embedding-level domain adaptation, referred to as feature-distribution adaptor (FDA) in [21]. It is shown as special case 8 in Tab. I.

In a way similar to Kaldi and CORAL+, FDA uses regularization to ensure uncertainty increase after the adaptation. The difference is that it is an embedding-level transformation in rather than a model-level transformation like that seen in Kaldi and CORAL+. It has been argued that embedding-level domain adaptation is more relevant than model-level domain adaptation because the frequently used embedding pre-processing techniques, such as linear discriminant analysis (LDA) for discriminant dimensionality reduction [8] and within-class covariance normalization (WCCN), rely on the parameters learned by using adapted embeddings.

Since OOD data's labels are available, we can further replace the empirical total covariance $\mathbf{C}_{\mathrm{O}}$ in (36) with PLDA total covariance $\mathbf{\Phi}_{\mathrm{tot,O}}$ from (31)

$$\begin{aligned}\phi_{\mathrm{I,pseudo}} =& \mathbf{\Phi}_{\mathrm{tot,O}}^{\frac{1}{2}}\mathbf{P}\hat{\Delta}^{\frac{1}{2}}\mathbf{P}^{\mathsf{T}}\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}\phi_{\mathrm{O}} \\ \mathbf{P}\hat{\Delta}\mathbf{P}^{\mathsf{T}} =& \Gamma_{max}(\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}\mathbf{C}_{\mathrm{I}}\mathbf{\Phi}_{\mathrm{tot,O}}^{-\frac{1}{2}}, \mathbf{I}).\end{aligned} \tag{37}$$

The corresponding modification of the PLDA between- and within speaker covariance matrices according to (25) are referred to as a modified Kaldi method (as special case 9 Kaldi* in Tab I and Sec VI), which was also introduced in [21] as well. This method is an extension from FDA to model-level adaptation. When no pre-processing is used before model training, the difference in results arising from the application of the two methods will only be that arising from the changes in the use of OOD total covariance calculated from the OOD data or the trained OOD model.

All of the above-mentioned methods are summarized as special cases in Tab. I. We show the relationships between them in Fig. 2 from the perspective of the three main factors of the generalized framework. For convenience, the relevant equations are shown in Tab. II.

VI. Experiments

Experiments were conducted on the recent NIST SRE'16, 18, and 19 CTS datasets [26] [43] [44]. Performance was evaluated in terms of equal error rate (EER) and minimum detection cost (minDCF).

TABLE III: Development data and evaluation data used in the experiments

Experiments	OOD data	InD data	Eval data
SRE16	MIXER	cmntrain unlabeled major	cmneval
SRE18	MIXER	cmn2eval	cmn2dev
SRE19	MIXER	cmn2dev	SRE'19 eval

A. Datasets

The latest SREs organized by NIST have focused on domain mismatch as a particular technical challenge. Switchboard, VoxCeleb 1 and 2, and MIXER corpora that consisted of SREs 04 – 06, 08, 10, and 12 were used to train an x-vector extractor. They were considered to be OOD data as they are English speech corpora, while SRE'16 is in Tagalog and Cantonese, and SRE'18 and 19 are in Tunisian Arabic.

Data augmentation was applied to the training set in the following ways: (a) Additive noise: each segment was mixed with one of the noise samples in the PRISM corpus [45] (SNR: 8, 15, or 20dB), (b) Reverberation: each segment was convolved with one of the room impulse responses in the REVERB challenge data [46], and (c) Speech encoding: each segment was encoded with AMR codec (6.7 or 4.75 kbps) [27].

Only the MIXER corpora and its augmentation, which consisted of 262,427 segments from 4,322 speakers in total, were used as OOD data in PLDA training. The development data and evaluation data used are shown in Tab. III. For SRE'16, only the *cmntrain unlabeled major* set (2,272 segments) was available as a development set. Thus, only unsupervised domain adaptation experiments were conducted on SRE'16. For the two partitions Tagalog and Cantonese in SRE'16, only equalized results are shown in this paper. SRE'18 has three datasets: an evaluation set (188 speakers, 13,451 segments), a development set (25 speakers, 1,741 segments), and an unlabeled set (2,332 segments). For SRE'19 there was no development set available, but it was collected together with SRE'18. Thus, they are considered to be in the same domain. Therefore, we followed the common benchmark setting and use the SRE'18 evaluation set as the development data for SRE'19 experiments. To use the same setting for SRE'18 experiments, we evaluated the development set. In this section, we refer to the evaluation dataset as InD data, and to the development set as enroll and test data, in order to avoid confusion. In the following experiments, we used SRE16, SRE18, and SRE19 to specifically present the experiments using the above-mentioned settings to differ from the datasets SRE'16, SRE'18 and SRE'19.

B. Experimental setup

To confirm the generalization of the proposed methods, we used two Time-delayed neural network structures for the x-vector extraction: a 43-layer TDNN (deep) and a 27-layer TDNN (shallow). Both have residual connections and a 2-head attentive statistics pooling, in the same way as in [27]. Additive margin softmax loss (s=40.0, m=0.15) was used for

TABLE IV: Performance of G-PLDA and HT-PLDA on SRE'16-19. LDA was not applied.

(a) EER(%)

	Shallow		Deep	
	G-PLDA	HT-PLDA	G-PLDA	HT-PLDA
SRE16 OOD	8.17	7.98	5.78	5.49
SRE18 OOD	8.02	7.21	6.37	5.15
SRE18 InD	7.49	5.59	4.39	3.93
SRE19 OOD	7.47	6.95	4.58	3.86
SRE19 InD	6.20	5.29	4.00	3.30

(b) $\mathrm{minDCF}_1/\mathrm{minDCF}_2$ using P_{Target} equivalent to 0.01 and 0.005, respectively

	Shallow		Deep	
	G-PLDA	HT-PLDA	G-PLDA	HT-PLDA
SRE16 OOD	0.662/0.762	0.560/0.621	0.462/0.539	0.433/0.491
SRE18 OOD	0.518/0.556	0.456/0.512	0.411/0.440	0.382/0.424
SRE18 InD	0.400/0.447	0.337/0.376	0.256/0.298	0.224/0.262
SRE19 OOD	0.538/0.601	0.513/0.577	0.364/0.424	0.329/0.395
SRE19 InD	0.501/0.567	0.420/0.477	0.313/0.369	0.258/0.302

the optimization [47]. The number of dimensions of the x-vector was 512. The mean shift was applied to OOD data using its mean. InD data and enroll and test data were centralized using InD data. LDA was used in some of the G-PLDA systems to reduce dimensionality to 150-dimension. The usage is clarified with the experiments later. In our interpolation domain adaptation experiments where LDA was applied, the LDA projection matrix was calculated from the training data for the base PLDA, $\mathbf{\Phi_0}$ in (11), and applied to the training data for all the PLDA in the interpolation, as well as the evaluation data. For the single InD G-PLDA and OOD G-PLDA training, LDA matrices were calculated from the InD data and OOD data, respectively. LDA is not used in any HT-PLDA systems. Our implementation of domain adaptation for HT-PLDA is based on [24], and λ is set as 2 as the default setting in the code[1].

We report here results in terms of equal error rate (EER) and the minimum of the normalized detection cost function (minDCF), for which we assume two prior target probabilities P_{Target} of 0.01 and 0.005, and equal weights of 1.0 between misses C_{Miss} and false alarms $C_{\mathrm{FalseAlarm}}$, as defined in the evaluation plan of NIST SRE' 16, 18, and 19 [26] [43] [44]. In the later experiments, we use, rather, $\mathrm{minC}_{\mathrm{primary}}$, which is averaged from the two minDCFs.

C. G-PLDA vs. HT-PLDA

We first evaluated the single G-PLDA and HT-PLDA trained from the mismatched (OOD) and matched (InD) data, using the two x-vector extractor structures. LDA was not applied. Tables IVa and IVb show performance using x-vectors extracted from the 27-layer TDNN (shallow) and 43-layer TDNN (deep), respectively. The two minDCF values were calculated using the two operating points in which P_{Target} equals to 0.01 and 0.005, respectively. They represent two points in the DET curve. A comparison of performance using HT-PLDA and G-PLDA within each of Tab. IVa and IVb shows that HT-PLDA outperformed G-PLDA by a large margin in EER and both minDCFs, for both OOD and InD domains. This effect remains the same when using either of the front-ends. Such observations are consistent with prior studies. This, again, demonstrates that it is essential to develop domain adaptation for HT-PLDA, which is the better baseline.

A comparison of the two tables shows that using a deeper neural network yields greater improvement in performance in G-PLDA systems then that in HT-PLDA systems. The average EER/$\mathrm{minC}_{\mathrm{primary}}$ reduction in G-PLDA systems is $33.0\%/29.4\%$, while it is $29.1\%/25.5\%$ in HT-PLDA systems.

The effectiveness of HT-PLDA can be seen to be larger in shallower networks than in deeper networks: $17.11\%/15.2\%$ and $13.6\%/11.5\%$ average reduction in EER/$\mathrm{minC}_{\mathrm{primary}}$ using the shallower network and deep network, respectively. Since the two minDCFs show consistent results, in the later experiments we instead used only $\mathrm{minC}_{\mathrm{primary}}$, which is averaged from the two. Since the systems with the deeper TDNN outperform those with the shallow one in all settings, we will next focus on and show only the experiments using the 43-layer TDNN (deep). The corresponding results for the 27-layer TDNN (shallow) are shown in Appendix B in [42].

TABLE V: Investigations of an LDA application to G-PLDA systems. The projection matrix is computed from raw x-vectors or adapted features; "*-1" indicates those whose LDA were calculated from the adapted features, while "*-2" represents those whose LDA were calculated from the raw x-vectors. X-vectors were extracted from the 43-layer network.

Results are shown as EER/$\mathrm{minC}_{\mathrm{primary}}$.

	SRE16	SRE18	SRE19
InD (no LDA)	- / -	4.39/0.277	4.00/0.341
OOD (no LDA)	5.78/0.500	6.37/0.426	4.58/0.394
InD	- / -	4.52/0.278	4.04/0.361
OOD	5.84/0.494	6.18/0.415	4.53/0.394
CORAL-1	5.67/0.410	5.15/0.279	4.90/0.390
CORAL-2	**4.03/0.353**	**4.31/0.251**	**3.89/0.316**
FDA-1	4.17/**0.331**	4.35/**0.228**	3.54/**0.288**
FDA-2	**3.76**/0.335	**4.22**/0.244	**3.50**/0.298

D. G-PLDA and HT-PLDA with domain adaptations

We evaluated the two PLDAs with embedding-level adaptation methods, i.e., CORAL+ and FDA, as noted in Section V. In G-PLDA, LDA and length normalization were applied to the x-vectors as pre-processing before PLDA evaluations. Thus, an LDA projection matrix can be calculated from either the raw x-vectors from which the CORAL transformation and FDA transformation were also derived, or from the transformed x-vectors after CORAL and FDA have been applied. We investigated the use of both LDA matrices (see Tab. V). It is worth noticing that LDA was applied to the OOD and InD PLDA systems here. Therefore, their performance differs from that in Tab. IV which is referred to as InD (no LDA) and OOD (no LDA) in Tab. V. Notably, in CORAL, we found that "CORAL-2", which used LDA calculated from the raw x-vectors, achieved the best performance in both measurements in the three data sets. For FDA, the advantage was shown

[1]https://github.com/bsxfan/meta-embeddings/tree/master/code/Niko/matlab/clean/VB4HTPLDA

TABLE VI: Domain adaptations in SRE16-SRE19 with two TDNN structured x-vector extractors. X-vectors were extracted from the 43-layer network. Results are shown as EER/$\text{minC}_{\text{primary}}$.

(a) SRE16

	G-PLDA	HT-PLDA
OOD	5.84/0.494	5.49/0.462
CORAL	4.03/0.353	5.64/0.470
FDA	**3.76/0.335**	**4.66**/0.426
CORAL+	4.01/0.350	4.77/0.449
Kaldi	4.04/0.342	4.96/**0.410**
Kaldi*	3.92/0.336	4.69/0.428

(b) SRE18 and SRE19. The weights in all the interpolations were chosen to be 0.5.

System	SRE18		SRE19	
	G-PLDA	HT-PLDA	G-PLDA	HT-PLDA
OOD	6.18/0.415	5.15/0.403	4.53/0.394	3.86/0.362
InD	4.52/0.278	3.93/0.243	4.04/0.361	3.30/0.280
CORAL	4.31/0.251	4.92/0.334	3.89/0.316	3.95/0.346
FDA	4.22/0.244	4.11/0.281	3.50/0.298	3.23/0.297
CORAL+	4.31/0.264	4.14/0.282	3.34/0.305	3.23/0.307
Kaldi [34]	4.20/0.271	4.23/0.280	3.82/0.315	3.77/0.350
Kaldi*	4.20/0.251	4.41/0.345	3.40/0.300	3.19/0.295
LIP(OOD)	3.85/0.222	3.41/0.222	3.12/0.286	2.56/0.239
LIP(CORAL)	3.89/0.189	3.36/0.198	3.19/0.291	2.47/**0.222**
LIP(FDA)	3.80/0.200	**3.29**/0.213	3.02/0.281	2.55/0.230
LIP(CORAL+)	3.68/0.212	3.41/0.201	2.90/0.275	2.46/0.228
LIP(Kaldi)	3.83/0.205	3.38/0.241	3.03/0.276	2.57/0.249
LIP(Kaldi*)	**3.64**/0.213	3.31/0.210	2.93/0.273	**2.42**/0.225
LIP-reg(OOD)	3.86/**0.195**	3.63/0.202	3.33/0.291	2.66/0.237
LIP-reg(CORAL)	3.77/0.199	3.38/**0.189**	3.08/0.279	2.55/0.223
LIP-reg(FDA)	4.36/0.210	3.43/0.201	**2.88/0.269**	2.42/0.225
LIP-reg(CORAL+)	3.78/**0.195**	3.58/0.194	3.08/0.283	2.58/0.232
LIP-reg(Kaldi)	3.73/0.220	3.36/0.230	3.13/0.280	2.61/0.244
LIP-reg(Kaldi*)	3.78/0.202	3.43/0.199	3.01/0.279	2.55/0.230

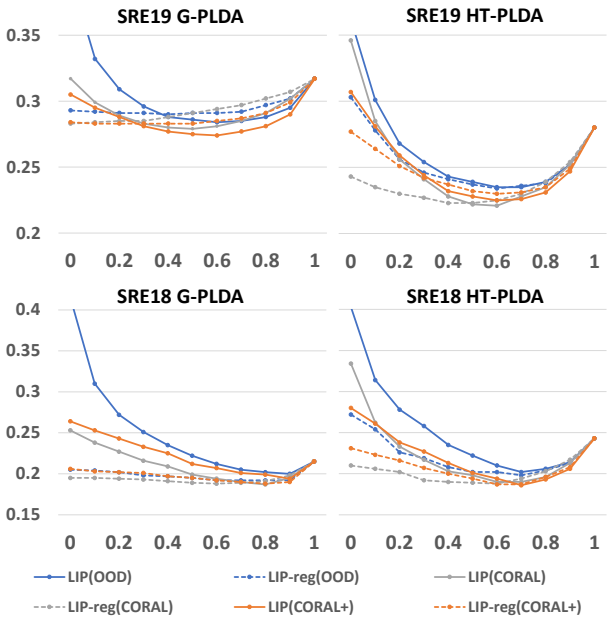

Fig. 4: $\text{minC}_{\text{primary}}$ of SRE18 and SRE19 evaluations using G-PLDA and HT-PLDA with varying interpolation weights. Three interpolation settings without and with regularizations are compared: (1) LIP between OOD and InD PLDA, (2) LIP between CORAL adapted PLDA and InD PLDA, and (3) LIP between CORAL+ adapted PLDA and InD PLDA. The regularizations use the InD PLDA as a reference. The x-vectors were extracted from the 43-layer network. X-axis shows the interpolation weight for InD PLDA, and the y-axis shows $\text{minC}_{\text{primary}}$.

only in EER, while minDCFs were slightly worse. For the later experiments of embedding-level domain adaptation for G-PLDA, we used LDA calculated from the raw x-vectors.

We show in Tab. VI several special cases of the generalized format of domain adaptation for the three datasets. Only unsupervised methods are shown for SRE16 evaluations. It first shows OOD PLDAs' performance results, for reference purposes, and five existing unsupervised domain adaptation methods' results, including those for CORAL, FDA, CORAL+, Kaldi adaptation, and the modified Kaldi. All five adaptation methods employed the correlation alignment factor of the generalized framework, and the latter four also employ the covariance regularization factor. The CORAL adaptation achieved improvements over those of the OOD results in the case of G-PLDA systems for all three datasets, while failing in working with HT-PLDA. The rest four adaptation methods significantly improved both G-PLDA and HT-PLDA systems for all three datasets. The results indicate the effectiveness of the covariance regularization factor, especially for HT-PLDAs. For SRE18 and SRE19 evaluations where the development data is available to train InD PLDA models, Tab. VIb shows that the G-PLDA with the above-mentioned unsupervised adaptation methods even outperformed InD G-PLDA system. For HT-PLDA, the application of FDA, CORAL+, and Kaldi* also outperformed the InD HT-PLDA in terms of EER for SRE19 evaluation, while a slight increase in minDCF. For the SRE18 evaluation, the application of the unsupervised adaptation methods did not bring improvement. To be noted, LDA was applied to all the G-PLDA systems in Tab. VI. Therefore, their baseline OOD and InD G-PLDA results are consistent with those in Tab. V rather than those in Tab. IV.

We next investigated the special cases specifically using the covariance matrix interpolation (LIP) factor of the generalized framework. Since InD PLDA outperformed OOD PLDA for both types of PLDA in the evaluations of both SRE18 and SRE19, we used it as the base PLDA and interpolated it with the above-mentioned unsupervised-adapted PLDA. As shown in Tab. VI, LIP and LIP-reg systems interpolate InD PLDA and the unsupervised-adapted PLDAs shown in the brackets in the "System" column. For LIP-reg systems, the InD PLDA is used as the reference PLDA for the regularization of the unsupervised-adapted PLDAs shown in the brackets before being interpolated. The weights in all the interpolations in Tab. VI were chosen to be 0.5. LDA projection matrix was calculated from the InD data and applied to all the LIP systems. We observed that all the LIP systems outperformed their single systems, and the observation is consistent for both G-PLDA and HT-PLDA and for both datasets. We conclude that the adaption by the linear interpolation with the InD PLDA further improved the performance over the unsupervised domain-adapted PLDAs.

On the basis of the LIP systems, we show their corresponding systems (LIP-reg) with a covariance regularization factor of the generalized framework. The InD PLDA was additionally used as the reference PLDA, $\mathbf{\Phi}_2$ in (11), for the regularization purpose. We observed that for both PLDAs, using regularization achieved similar performance to that without regularization. Above is the analysis using 0.5 as the interpolation weight, and next, we looked into different weights.

E. Robustness of domain adaptations using regularization

To further confirm the effectiveness of the regularization factor in the linear fusions on speaker verification performance, we further investigated the robustness of the performance against varying interpolation weights. We take the next three settings as examples: LIP between OOD and InD PLDA, LIP between CORAL adapted PLDA and InD PLDA, LIP between CORAL+ adapted PLDA and InD PLDA, and these three settings with regularizations using InD PLDA as a reference. The plots in Fig. 4 show that although in the same settings the optimal $\mathrm{minC}_{\mathrm{primary}}$ may be better when not using regularization, those with regularization generally were less sensitive to the interpolation weights. We observe in all the four plots that a bigger range of interpolation weights achieved lower $\mathrm{minC}_{\mathrm{primary}}$ when regularization was applied. Specifically, with the varying interpolation weights from 0 to 1, the minDCFs have the range 0.223–0.333 on average over all the LIP systems in Fig. 4, while the range for LIP-reg systems is smaller, i.e., 0.224–0.268, with a lower maximum minDCF as 0.268. The average minDCF of the LIP-reg systems is 0.237, lower than that of the LIP systems at 0.253. The standard deviation of the LIP-reg systems is 0.013, much smaller than that of the LIP systems at 0.032. Thus, we conclude that the regularization improved robustness for interpolations w.r.t. varying interpolation weights. This was constant in both G-PLDA and HT-PLDA experiments.

VII. CONCLUSION

We have proposed here a generalized framework for domain adaptation of PLDA in speaker recognition that works with both unsupervised and supervised methods, as well as two new techniques: (1) correlation-alignment-based interpolation and (2) covariance regularization. The generalized framework enables us to combine the two techniques and also several existing supervised and unsupervised domain adaptation methods into a single formulation. Further, our proposed regularization technique ensures robustness for interpolations w.r.t. varying interpolation weights, which in practice is essential. In future, we intend to evaluate this approach's effectiveness on other DNN-based speaker embeddings.

REFERENCES

[1] J. H. L. Hansen and T. Hasan, "How humans and machines recognize voices: a tutorial review," in *IEEE Signal Processing Magazine*, vol. 32, 2015, p. 74–99.

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

[3] E. Variani, X. Lei, E. McDermott, I. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018.

[5] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018.

[6] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," in *Proc. IEEE Transactions on Audio, Speech and Language Processing*, 2011.

[7] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. Interspeech*, 2006.

[8] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[9] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007.

[10] S. Ioffe, "Probabilistic linear discriminant analysis," in *Proc. ECCV*, 2006.

[11] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey*, 2010.

[12] H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, "Speaker augmentation and bandwidth extension for deep speaker embedding," in *Proc. Interspeech*, 2019.

[13] Q. Wang, K. A. Lee, and T. Liu, "Scoring of large-margin embeddings for speaker verification: Cosine or PLDA?" in *Proc. Interspeech*, 2022.

[14] Q. Wang, W. Rao, P. Guo, and L. Xie, "Adversarial training for multi-domain speaker recognition," in *12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021.

[15] Y. Wei, J. Du, H. Liu, and Z. Zhang, "CentriForce: Multiple-domain adaptation for domain-invariant speaker representation learning," in *IEEE Signal Processing Letters*, 2022, pp. 807–811.

[16] H. Zhang, L. Wang, K. A. Lee, M. Liu, J. Dang, and H. Meng, "Meta-generalization for domain-invariant speaker verification," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, 2023, pp. 1024–1036.

[17] A. Misra and J. Hansen, "Spoken language mismatch in speaker verification: an investigation with NIST-SRE and CRSS Bi-ling Corpora," in *Proc. IEEE SLT*, 2014.

[18] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. ICASSP*, 2014.

[19] J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Proc. Odyssey*, 2018.

[20] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of plda," in *Proc. ICASSP*, 2019.

[21] P.-M. Bousquet and M. Rouvier, "On robustness of unsupervised domain adaptation for speaker recognition," in *Interspeech*, 2019.

[22] K. A. Lee, H. Yamamoto, K. Okabe, Q. Wang, L. Guo, T. Koshinaka, J. Zhang, and K. Shinoda, "NEC-TT system for Mixed-Bandwidth and Multi-Domain speaker recognition," in *Computer Speech & Language*, 2020.

[23] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011.

[24] A. Silnova, N. Brummer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast variational bayes for heavy-tailed plda applied to i-vectors and x-vectors," in *Proc. Interspeech*, 2018, pp. 72–76.

[25] N. Brummer, A. Silnova, L. Burget, and T. Stafylakis, "Gaussian meta-embeddings for efficient scoring of a heavy-tailed plda model," in *Proc. Odyssey*, 2018.

[26] National Institute of Standards and Technology, "NIST 2018 Speaker Recognition Evaluation Plan," *NIST SRE*, 2018.

[27] K. A. Lee, H. Yamamoto, K. Okabe, Q. Wang, L. Guo *et al.*, "The NEC-TT 2018 speaker verification system," in *Proc. Interspeech*, 2019.

[28] P. Matejka, O. Plchot, O. Glembek, L. Burget, J. Rohdin *et al.*, "13 years of speaker recognition research at BUT, with longitudinal analysis of NIST SRE," in *Computer Speech & Language*, vol. 63, 2020, p. 101035.

[29] Q. Wang, K. Okabe, K. A. Lee, and T. Koshinaka, "A generalized framework for domain adaptation of PLDA in speaker recognition," in *Proc. ICASSP*, 2020.

[30] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proc. ICASSP*, 2014.

[31] R. Li, W. Zhang, and D. Chen, "The CORAL++ algorithm for unsupervised domain adaptation of speaker recognition," in *Proc. IEEE ICASSP*, 2022.

[32] T. Chen and E. Khoury, "Speaker embedding conversion for backward and cross-channel compatibility," in *Proc. IEEE ICASSP*, 2022.

[33] 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*, 2014.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2011.

[35] Q. Wang, H. Yamamoto, and T. Koshinaka, "Domain adaptation using maximum likelihood linear transformation for PLDA-based speaker verification," in *Proc. ICASSP*, 2016.

[36] P. Matejka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *Proc. IEEE ICASSP*, 2011.

[37] K. A. Lee, Q. Wang, and T. Koshinaka, "Xi-vector embedding for speaker recognition," in *IEEE Signal Processing Letters*, 2021.

[38] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," in *IEEE Trans. Audio, Speech and Language Processing*, 2008.

[39] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, 2016.

[40] R. A. Horn and C. R. Johnson, "Matrix analysis (2nd ed.)," in *UK: Cambridge University Press*. Cambridge, 2013.

[41] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," in *The American Statistician*, 2018, pp. 1–6.

[42] Q. Wang, K. Okabe, K. A. Lee, and T. Koshinaka, "Generalized domain adaptation framework for parametric back-end in speaker recognition," in *https://arxiv.org/abs/2305.15567*, 2023.

[43] National Institute of Standards and Technology, "NIST 2016 Speaker Recognition Evaluation Plan," *NIST SRE*, 2016.

[44] National Institute of Standards and Technology, "NIST 2019 Speaker Recognition Evaluation: CTS Challenge," in *NIST SRE*, 2019.

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

[46] K. Kinoshita, M. Delcroix, S. Gannot, E. A. Habets, R. HaebUmbach, W. Kellermann, V. Leutnant, R. Maas, T. Nakatani, B. Raj, and et al., "A summary of the reverb challenge: state-ofthe-art and remaining challenges in reverberant speech processing research," in *URASIP Journal on Advances in Signal Processing*, 2016.

[47] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," in *Proc. IEEE Signal Processing Letters*, 2018.

Qiongqiong Wang is currently a Senior Research Engineer at the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore. She was a researcher at the Biometrics Research Laboratories, NEC Corporation, Japan, from 2013 to 2021. She received her M.E in computer science from the Tokyo Institute of Technology, Japan, in 2013, and B.E from the Undergraduate School of Physics, Shanghai Jiao Tong University, China, in 2011. Her research focuses on speaker recognition, deception detection, speech anti-spoofing, and speech enhancement.

Koji Okabe is currently a Principal Researcher at the Data Science Laboratories, NEC Corporation, Japan. He works in research and development of speech recognition technology. He received his master's degree from the University of Tokyo, Japan, in 2007. His research focuses on automatic speech recognition and speaker recognition.

Kong Aik Lee (Senior Member, IEEE) is currently an Associate Professor with Singapore Institute of Technology, Singapore. He holds a concurrent appointment as a Senior Scientist and a Group Leader at the Agency for Science, Technology and Research (A*STAR), Singapore. He was a Senior Principal Researcher at the Data Science Research Laboratories, NEC Corporation, Japan, from 2018 to 2020. He received his Ph.D. degree from Nanyang Technological University, Singapore, in 2006. After which he joined the Institute for Infocomm Research, Singapore, as a Research Scientist and then a Strategic Planning Manager (concurrent appointment). He was the recipient of the Singapore IES Prestigious Engineering Achievement Award 2013 for his contribution to voice biometrics technology, and the Outstanding Service Award by IEEE ICME 2020. Currently, he serves as an Editorial Board Member for Elsevier Computer Speech and Language (2016 - present) and was an Associate Editor for IEEE/ACM Transactions on Audio, Speech, and Language Processing (2017 - 2021). He is an elected member of the IEEE Speech and Language Processing Technical Committee and was the General Chair of the Speaker Odyssey 2020 Workshop. His research focuses on the automatic and paralinguistic analysis of speaker characteristics, ranging from speaker recognition, language and accent recognition, diarization, voice biometrics, spoofing, and countermeasure.

Takafumi Koshinaka (Member, IEEE) received his B.E. and M.E. in aeronautical engineering from Kyoto University, in, respectively, 1991 and 1993. In 1993, he joined the NEC Corporation and had been serving as a Senior Principal Researcher from 2013 to 2020. He received his Ph.D. in computer science from the Tokyo Institute of Technology in 2013. He is currently a Professor with the School of Data Science, Yokohama City University. His current research interests include pattern recognition and machine learning for speech and image recognition and understanding as well as for natural language processing. He was a Board Member (2017 – 2019) of the Japanese Society for Artificial Intelligence (JSAI) and a General Co-chair of the Speaker Odyssey 2020 Workshop. He is a Representative (2019 – present) of JSAI as well as a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Association for Natural Language Processing (ANLP), and the Acoustical Society of Japan (ASJ).