Variational Domain Adversarial Learning with Mutual Information Maximization for Speaker Verification

Youzhi Tu, Man-Wai Mak, *Senior Member, IEEE,* and Jen-Tzung Chien, *Senior Member, IEEE*

***Abstract*—Domain mismatch is a common problem in speaker verification (SV) and often causes performance degradation. For the system relying on the Gaussian PLDA backend to suppress the channel variability, the performance would be further limited if there is no Gaussianity constraint on the learned embeddings. This paper proposes an information-maximized variational domain adversarial neural network (InfoVDANN) that incorporates an InfoVAE into domain adversarial training (DAT) to reduce domain mismatch and simultaneously meet the Gaussianity requirement of the PLDA backend. Specifically, DAT is applied to produce speaker discriminative and domain-invariant features, while the InfoVAE performs variational regularization on the embedded features so that they follow a Gaussian distribution. Another benefit of the InfoVAE is that it avoids posterior collapse in VAEs by preserving the mutual information between the embedded features and the training set so that *extra* speaker information can be retained in the features. Experiments on both SRE16 and SRE18-CMN2 show that the InfoVDANN outperforms the recent VDANN, which suggests that increasing the mutual information between the embedded features and input features enables the InfoVDANN to extract *extra* speaker information that is otherwise not possible.**

***Index Terms*—Speaker verification, domain adaptation, domain adversarial training, variational autoencoder, mutual information.**

I. Introduction

THE objective of speaker verification (SV) is to determine whether the identity of a test utterance matches that of a target speaker. To achieve optimal performance, SV systems rely on the condition that the training data (or source-domain data) share the same distribution with the test data (or target-domain data). In practice, however, this condition can hardly be met and domain mismatch occurs, which poses a great challenge to SV. As a result, it is necessary to adapt the trained models based on some target-domain data. This strategy is known as domain adaptation (DA), which is a branch of transfer learning. However, due to the high cost of data labeling, usually only a small amount of labeled data or even no labeled data from the target domain are available. This difficulty motivates researchers to seek advanced DA methods to alleviate the domain mismatch problems.

A. Related Work

Early DA methods are implemented in a supervised manner, which require speaker labels from the target domain [1], [2]. Recently, research on DA has been focusing on the semi-supervised learning scenario where only some unlabeled target-domain data are available besides a large amount of labeled source-domain data. One approach is to hypothesize the speaker labels through unsupervised learning. With the hypothesized labels, one can adapt the source-domain probabilistic linear discriminant analysis (PLDA) model [3] to the target domain [4]–[6].

Another category aims to learn a domain-invariant space for transforming the source-domain i-vectors [7] so that the PLDA models trained on these transformed i-vectors can match the target-domain data. Such methods include inter-dataset variability compensation [8], dataset-invariant covariance normalization [9], and correlation alignment (CORAL) [10]. Although CORAL is as simple as aligning the second-order statistics of the source and target domains, it can achieve comparable performance as the competitive Kaldi's PLDA adaptation[1] in SRE16. Recently, a feature-level method called feature-Distribution Adaptor [11] was proposed to mitigate the adverse effect of inaccurate information estimated from the limited in-domain data. Yet we may directly adapt the parameters of PLDA models, e.g., CORAL+ [12] aligns the covariance matrices in PLDA models for DA.

In addition to explicitly finding a transformation matrix in the feature level [8]–[11] or directly adapting PLDA parameters in the model level [12], DNN-based DA has also been applied to learn a domain-invariant space. In [13], autoencoder-based DA was proposed to reduce channel mismatch. In [14]–[16], Lin *et al.* applied maximum mean discrepancy (MMD) [17] as a distribution distance metric and produced the features that are less domain-dependent. Since the emergence of generative adversarial networks (GAN) [18], adversarial learning has been applied to unsupervised DA [19]–[24]. In [22], Wang *et al.* utilized domain adversarial training (DAT) [19] to generate speaker discriminative and domain-invariant representations, which outperformed traditional DA approaches in the Domain Adaptation Challenge 2013. Rohdin

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 152137/17E, and Taiwan MOST, Grant No. 109-2634-F-009-024.

Y. Tu and M. W. Mak are with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (email: 918tyz@gmail.com; enmwmak@polyu.edu.hk). J. T. Chien is with the Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan (email: jtchien@nctu.edu.tw).

This paper has supplementary downloadable material at https://github.com/enmwmak/InfoVDANN, provided by the author. Contact enmwmak@polyu.edu.hk for further questions about this work.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

et al. [23] implemented DAT in an end-to-end fashion to produce features that are invariant to languages.

Although adversarial-learning based unsupervised DA has greatly boosted the SV performance under domain mismatch scenarios, it may lead to non-Gaussian latent vectors, which do not meet the Gaussianity requirement of the PLDA backend. This problem can be solved by using the heavy-tailed PLDA [25], [26] or applying the i-vector length normalization [27]. However, the former is more computationally expensive than the Gaussian PLDA and the latter is not really a Gaussianization procedure but a sub-optimal compromise. Recently, there have been some works trying to Gaussianize speaker embeddings obtained by neural networks. In [28], Gaussian-constrained training was proposed by incorporating an l_2-regularizer into the cross-entropy loss. Chien *et al.* [29] integrated a Gaussianity constraint by using a variational autoencoder (VAE) [30] in training a GAN for data augmentation.

Using VAEs for regularization is effective in Gaussianizing the speaker embeddings [29], [31]. However, training VAEs by maximizing the evidence lower bound (ELBO) of log likelihood has some problems, which can cause failure in learning informative latent representations [32]–[36]. One problem is that given a finite amount of training data, a VAE tends to overfit the training data and generate inaccurate variational posteriors [37], [38]. Another problem is that if the decoder is flexible enough, e.g., it uses a recurrent neural network (RNN) as in [39] or it is characterized by an arbitrarily complex distribution constructed by a normalizing flow [40], [41], a VAE can produce noninformative latent vectors independent of the inputs. This problem is referred to as posterior collapse [35], [42]–[44]. This is undesirable because our objective is to learn the meaningful representations.

Several methods have been proposed to address the posterior collapse problem, e.g., applying the Kullback-Leibler (KL) cost annealing [39], using a variational mixture of posteriors in the prior of latent features [37], reducing the amortization gap [43], skipping the connections in the decoder [44], aggressively training the encoder within an update of the decoder [35], etc. In [45], self attention was introduced to alleviate posterior collapse by attending the latent information when training variational recurrent autoencoders. Some advanced flow-based generative models [46], [47] could also be applied to deal with the posterior collapse by relaxing the strict assumption that the prior follows a standard Gaussian distribution and using more complex variational posteriors.

B. Main Idea of This Work

In [48], Tu *et al.* proposed a variational domain adversarial neural network (VDANN) by incorporating a VAE into the standard DANN [19] to regularize the distribution of the embedded features. The authors demonstrated that the transformed embeddings were more Gaussian than the DANN-transformed ones, which led to performance improvement.

However, as discussed in Section I-A, the VAE may not learn any useful embeddings even though the ELBO has been maximized. In [36], Zhao *et al.* proposed the InfoVAE, a variant of VAE with the information-maximized latent representation, to address the problems in VAEs. The idea is to increase the contribution of the KL divergence between the aggregated variational posterior [33], [49] and the latent prior so that the latent inference and data reconstruction can be balanced. Also, the dependence of the latent vectors on the inputs can be enhanced by explicitly incorporating a mutual information (MI) term into the objective function.

In this paper, we adopt the idea of InfoVAE and extend the VDANN [48] for unsupervised DA. With the InfoVAE, the learned features can sufficiently reflect the meaningful information from the inputs, while simultaneously retain the benefit of VDANN to produce speaker discriminative, domain-invariant and Gaussian distributed features. We refer to the resulting information-maximized variational domain adversarial neural network as InfoVDANN. InfoVDANN improves VDANN in two aspects: 1) it balances the latent representation learning and data reconstruction to avoid overfitting; 2) it preserves the MI between the learned embeddings and the inputs to improve speaker discriminative capacity.

This paper is organized as follows. In Section II, we introduce variational domain adversarial learning and clarify the novelty of this work compared with previous works. Section III details the principle of the proposed InfoVDANN and its two variants: MMD–VDANN and AAE–VDANN. The experimental settings and results are reported in Section IV and Section V, respectively. We then give concluding remarks in Section VI.

II. Variational Domain Adversarial Learning

This study works on SV by using the variational domain adversarial learning. The fundamentals of variational autoencoders (VAEs) and domain adversarial neural networks (DANNs) are addressed.

A. Variational Autoencoder

Suppose we have a training set $\mathcal{X}$ whose true data distribution is denoted as $p_{\mathcal{D}}(\mathbf{x})$ ($\mathbf{x} \in \mathcal{X}$) and its underlying generation is determined by a latent variable set $\mathcal{Z}$. A VAE can be optimized by maximizing the evidence lower bound (ELBO) of log likelihood [30], [36]

$$\begin{aligned}\text{ELBO} = \ & \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[- \text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)\right. \\ & \left. + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right] \\ \propto & - \text{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right) \\ & - \mathbb{E}_{q_\phi(\mathbf{z})}\left[\text{KL}\left(q_\phi(\mathbf{x}|\mathbf{z}) \| p_\theta(\mathbf{x}|\mathbf{z})\right)\right], \end{aligned} \tag{1}$$

where ϕ and θ are parameters of the encoder and decoder, respectively, $q_\phi(\mathbf{z}|\mathbf{x})$ is an approximation of the intractable true posterior $p_\theta(\mathbf{z}|\mathbf{x})$, and $p(\mathbf{z})$ is the prior of $\mathbf{z}$ which is generally a standard Gaussian $\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$. In (1), $q_\phi(\mathbf{z})$ is the aggregated posterior [33], [49]: $q_\phi(\mathbf{z}) = \int_{\mathbf{x}} p_{\mathcal{D}}(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{x}$. Because $q_\phi(\mathbf{z})$ requires an aggregation over the entire training set $\mathcal{X}$, it cannot be computed exactly. In practice, we can approximate $q_\phi(\mathbf{z})$ by a Monte Carlo estimate [50], [51].

Maximizing the ELBO directly can lead to some problems. First, due to the inherent properties of the ELBO, maximization can lead to very inaccurate variational posteriors even though the ELBO can be maximized to infinity [36]. This limitation is exacerbated when the dimension of the latent variables is much lower than the input dimension; in such situation, optimization tends to sacrifice variational inference to enhance data reconstruction. This is because according to (1), if the dimension of $\mathbf{x}$ is much higher than that of $\mathbf{z}$, maximization of the ELBO will emphasize the second term, i.e., data reconstruction. This bias in emphasis can easily cause overfitting. Second, if the decoder is flexible enough, VAE training will ignore the information between the latent features and the inputs, leading to noninformative representations [36], [42]. This issue is known as the posterior collapse [35], [43], [44] in which the learned features will not depend on the training data. In [36], InfoVAE was proposed to address both problems.

B. Domain Adversarial Neural Network

DANN [19] aims to learn a domain-invariant latent space by adversarial training for unsupervised domain adaptation. A standard DANN consists of three subnetworks: a feature extractor E, a label predictor C, and a domain discriminator D, where both C and D take the output of E as the input. Their parameters are denoted by ϕ_e, θ_c and θ_d, respectively. Given a source domain set $\mathcal{X}^S = \{\mathbf{x}_1^S, \dots, \mathbf{x}_{N_S}^S\}$ and a target domain set $\mathcal{X}^T = \{\mathbf{x}_1^T, \dots, \mathbf{x}_{N_T}^T\}$, where N_S and N_T are the number of samples in $\mathcal{X}^S$ and $\mathcal{X}^T$, respectively. Denote $\mathbf{y} = \{y_i\}$ corresponding to $\mathcal{X}^S$ as the one-hot main task labels and $\mathbf{d} = \{d_i\}$ corresponding to $\{\mathcal{X}^S, \mathcal{X}^T\}$ as the domain labels, respectively. Define the loss function of DANN as

$$\begin{aligned}\mathcal{L}(\theta_c, \theta_d, \phi_e) &= \mathcal{L}_C(\theta_c, \phi_e) - \alpha\mathcal{L}_D(\theta_d, \phi_e) \\ &= \sum_{\mathbf{x}_i \in \mathcal{X}^S} \mathcal{L}_C\left(C(E(\mathbf{x}_i), y_i)\right) \\ &\quad - \alpha \sum_{\mathbf{x}_i \in \{\mathcal{X}^S,\, \mathcal{X}^T\}} \mathcal{L}_D\left(D(E(\mathbf{x}_i), d_i)\right), \end{aligned} \tag{2}$$

where $\mathcal{L}_C(\cdot)$ and $\mathcal{L}_D(\cdot)$ are the loss functions for C and D, respectively. α weights the domain discrimination loss during training. The minmax optimization in DANN is denoted as follows

$$\min_{\theta_c, \phi_e} \max_{\theta_d} \mathcal{L}(\theta_c, \theta_d, \phi_e). \tag{3}$$

After training, the features encoded by the extractor are not only task discriminative but also domain-invariant. This feature extractor is used to calculate embeddings for later tasks.

C. Relation to Previous Works

One common characteristic of [8]–[12] is that they performed DA *without* using DNNs. For DNN-based DA methods, [13] and [14]–[16] applied Euclidean distance and MMD to measure the discrepancy between different distributions, respectively. Different from these distance metrics, [22] and [23] used adversarial training to learn domain-invariant representations. But because there is no constraint on the latent features learned by DANN, the adversarial training may lead to non-Gaussian latent vectors, which would break the assumption of the Gaussian PLDA backend. VDANN overcame this limitation by regularizing the learned embeddings using a VAE in DAT so that they were Gaussian distributed. Thus, VDANN differs from DANN in this *variational* regularization.

However, a potential limitation of the VDANN is that posterior collapse may occur while training the VAE, leading non-informative speaker representations. The proposed InfoVDANN follows the framework of *variational* DAT in that it performs domain adaptation and Gaussianity regularization simultaneously. However, a major difference with the previous VDANN is that it addresses the posterior collapse problem by explicitly incorporating an MI term in the loss function. Maximizing this term enables the InfoVDANN to preserve more speaker information into the learned embeddings, which is the novel part of the method.

In particular, MMD–VDANN and AAE–VDANN are proposed in Section III-C as two specialized variants of the InfoVDANN to leverage MI. The difference between MMD–VDANN and AAE–VDANN is that MMD–VDANN minimizes the MMD between the aggregated posterior $q_\phi(\mathbf{z})$ and the prior $p(\mathbf{z})$ as a proxy to minimize the generalized divergence $\mathrm{D}_g\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right)$, whereas AAE–VDANN applies adversarial training to minimize this divergence. We propose these two variants to demonstrate that the performance of InfoVDANN is *not* sensitive to how the generalized divergence is minimized. In other words, InfoVDANN would *not* be biased towards a specific divergence between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. We will verify this through the experimental results in Section V.

From [48], we see that DANN is a special case of the VDANN. By removing the sampling operation and the decoder in the VAE, VDANN becomes the DANN. In Section III-B, we will show that both the VDANN and DANN are special cases of the InfoVDANN: InfoVDANN becomes the VDANN if the MI term is removed from the objective function, and it becomes the DANN if we further remove the sampling operation and the decoder in the VAE.

III. Information-Maximized Variational Domain Adversarial Neural Network

This study pursues an informative latent variable model where the MI between the input and the latent variables is maximized. The solution based on information-maximized adversarial learning is accordingly proposed.

A. Information-Maximized VAE

In [36], a new objective function was proposed based on (1) to address the problems in VAEs. The objective includes 1) adding a scalar to increase the contribution of $\mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right)$ and to counteract the dimension imbalance between $\mathcal{X}$ and $\mathcal{Z}$ and 2) incorporating an MI term that explicitly retains high mutual information between $\mathbf{x}$ and $\mathbf{z}$. The resulting model is called InfoVAE whose objective is

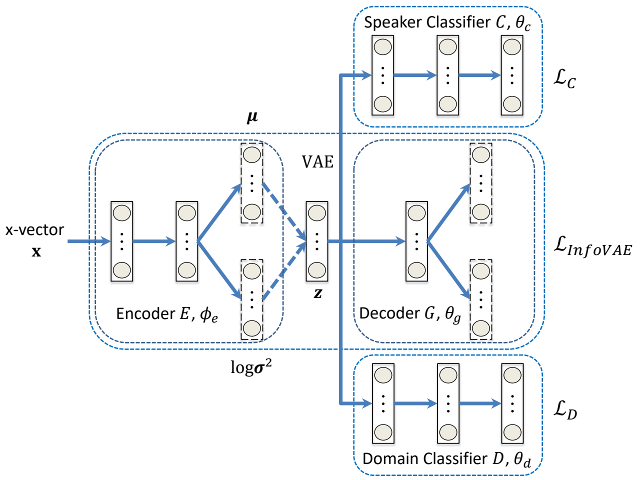


Fig. 1. Schematic of InfoVDANN. The solid and dashed arrows represent network connections and stochastic sampling, respectively. After training, the transformed features are extracted from the $\mathbf{z}$ nodes.

expressed as follows:

$$\begin{aligned}\mathrm{ELBO}_{\mathrm{InfoVAE}} &= -\lambda \mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right) + \eta I_q(\mathbf{x};\mathbf{z}) \\ &\quad - \mathbb{E}_{q_\phi(\mathbf{z})}\left[\mathrm{KL}\left(q_\phi(\mathbf{x}|\mathbf{z}) \| p_\theta(\mathbf{x}|\mathbf{z})\right)\right] \qquad (4) \\ &\propto \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \\ &\quad - (1-\eta)\mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[\mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)\right] \\ &\quad - (\lambda - 1 + \eta)\mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right), \qquad (5)\end{aligned}$$

where $I_q(\mathbf{x};\mathbf{z})$ is the MI between $\mathbf{x}$ and $\mathbf{z}$ under $q_\phi(\mathbf{x},\mathbf{z})$. The hyperparameter λ compensates for the dimension imbalance between $\mathbf{x}$ and $\mathbf{z}$, so that the variational inference and data reconstruction can be balanced. η signifies the importance of maintaining high mutual information between the original and latent vectors. We rewrite (4) as (5) because the MI term is difficult to compute directly, especially in high-dimensional spaces [52]. Note that we can further generalize the $\mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right)$ in (5) to broader divergence families for efficient optimization, e.g., we may use MMD [17] as a divergence measure or introduce a discriminator and apply adversarial training to distinguish samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$, similar to the adversarial autoencoder (AAE) [49].

B. Information-Maximized VDANN

One desirable property of VAE is that the term $\mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)$ in (1) can be considered as a regularizer that constrains the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to be close to the desired prior $p(\mathbf{z})$. Therefore, if we constrain $p(\mathbf{z})$ to be a multivariate Gaussian distribution, the encoder will likely to produce Gasussian latent vectors, which is amenable to PLDA modeling. This is also the key idea of the Gaussian regularization in the VDANN.

Due to the problems in training VAEs described in Section III-A, we propose an InfoVDANN by incorporating an InfoVAE into the DANN [19] to learn features that can sufficiently characterize the latent information from the inputs while simultaneously leverage the benefit of the VDANN.

As shown in Fig. 1, the InfoVDANN has a similar structure as the VDANN. It consists of a speaker predictor C, a domain classifier D, and a VAE that comprises an encoder E and a decoder G. The network parameters are denoted as θ_c, θ_d, ϕ_e, and θ_g, respectively.

Suppose the training set $\mathcal{X} = \{\mathcal{X}^r\}_{r=1}^R$ comprises N samples from R domains, where $\mathcal{X}^r = \{\mathbf{x}_1^r, \ldots, \mathbf{x}_{N_r}^r\}$ contains N_r samples from the r-th domain. Also we denote $\mathbf{y} = \{y_{ik}\}$ and $\mathbf{d} = \{d_i\}$ as the one-hot speaker labels and domain labels, respectively. The total number of training speakers is set to K.

Assume that $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, and for a given $\mathbf{x}_i$, the true posterior also follows a Gaussian distribution with mean vector $\boldsymbol{\mu}_i$ and a diagonal covariance matrix $\mathrm{diag}\left(\boldsymbol{\sigma}_i^2\right)$, i.e., $q_\phi(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}\left(\mathbf{z}; \boldsymbol{\mu}_i, \mathrm{diag}\left(\boldsymbol{\sigma}_i^2\right)\right)$. Applying the reparameterization trick in sampling $\mathbf{z}$'s from $q_\phi(\mathbf{z}|\mathbf{x}_i)$, we obtain the l-th latent sample $\mathbf{z}_{il} = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_l$, where $\boldsymbol{\epsilon}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is the Hadamard product. Substitute these terms into the *negative* of (5), we obtain the InfoVAE loss function

$$\begin{aligned}\mathcal{L}_{\mathrm{InfoVAE}}\left(\theta_g, \phi_e\right) &= -\frac{1}{N}\sum_{r=1}^{R}\sum_{i=1}^{N_r}\left\{\frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(\mathbf{x}_i^r|\mathbf{z}_{il}^r\right)\right. \\ &\quad - \frac{1-\eta}{2}\sum_{j=1}^{J}\left[\left(\mu_{ij}^r\right)^2 + \left(\sigma_{ij}^r\right)^2 - 1 - \log\left(\sigma_{ij}^r\right)^2\right] \\ &\quad \left. - (\lambda - 1 + \eta)\frac{1}{L}\sum_{l=1}^{L}\left[\log q_\phi(\mathbf{z}_{il}^r) - \log p(\mathbf{z}_{il}^r)\right]\right\}, \qquad (6)\end{aligned}$$

where J is the dimension of $\mathbf{z}$ and L denotes the number of sampled latent variables for a given $\mathbf{x}$. The hyperparameters λ and η are consistent with those in (5). The first term on the right-hand side of (6) is the data reconstruction error, whereas the second term is the analytical expression of $\mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right)$. The third term is the Monte Carlo estimate of $\mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right)$ in (5), i.e.,

$$\mathrm{KL}\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log q_\phi(\mathbf{z}) - \log p(\mathbf{z})\right]. \qquad (7)$$

In practice, we set $L = 1$ as suggested in [30].

To train the InfoVDANN, we define the loss of InfoVDANN as

$$\begin{aligned}\mathcal{L}_{\mathrm{InfoVDANN}}(\theta_c, \theta_d, \phi_e, \theta_g) &= \mathcal{L}_C(\theta_c, \phi_e) - \alpha\mathcal{L}_D(\theta_d, \phi_e) \\ &\quad + \beta\mathcal{L}_{\mathrm{InfoVAE}}(\phi_e, \theta_g), \qquad (8)\end{aligned}$$

where

$$\mathcal{L}_C(\theta_c, \phi_e) = \frac{1}{N}\sum_{r=1}^{R}\sum_{i=1}^{N_r}\left\{-\sum_{k=1}^{K} y_{ik}^r \log C\left(E\left(\mathbf{x}_i^r\right)\right)_k\right\} \qquad (9)$$

and

$$\mathcal{L}_D(\theta_d, \phi_e) = \frac{1}{N}\sum_{r=1}^{R}\sum_{i=1}^{N_r}\left\{-d_i^r \log D\left(E\left(\mathbf{x}_i^r\right)\right)_r\right\} \qquad (10)$$

are the categorical cross-entropy losses for speaker predictor and domain classifier, respectively; $\mathcal{L}_{\mathrm{InfoVAE}}(\phi_e, \theta_g)$ is defined as (6). The subscript k in (9) indexes the speakers. The hyperparameters α and β control the contribution of $\mathcal{L}_C$ and $\mathcal{L}_{\mathrm{InfoVAE}}$, respectively.

During training, for each mini-batch, we first optimize D by minimizing $\mathcal{L}_D(\theta_d, \phi_e)$ with respect to θ_d. θ_d is then fixed while training the remaining parts of the InfoVDANN.

Through adversarial training, the InfoVDANN learns a domain-invariant space. Specifically, applying adversarial training on E while keeping θ_d fixed together with minimizing the cross-entropy loss of C with respect to ϕ_e will make E to produce a domain-invariant but speaker discriminative representation through $\mathbf{z}$ in Fig. 1. Simultaneously, minimizing the InfoVAE loss will regularize the learned representations to be Gaussian. Also this minimization will feed more meaningful speaker information into the latent features through maximizing the MI between $\mathbf{x}$ and $\mathbf{z}$. The minimax optimization can be summarized as follows:

$$\min_{\theta_c,\phi_e,\theta_g}\max_{\theta_d}\mathcal{L}_{\text{InfoVDANN}}(\theta_c,\theta_d,\phi_e,\theta_g). \tag{11}$$

Alternatively, (11) can be expressed as

$$\hat{\theta}_d = \operatorname*{argmax}_{\theta_d}\mathcal{L}_{\text{InfoVDANN}}(\hat{\theta}_c,\theta_d,\hat{\phi}_e,\hat{\theta}_g), \tag{12}$$

$$\left(\hat{\theta}_c,\hat{\phi}_e,\hat{\theta}_g\right) = \operatorname*{argmin}_{\theta_c,\phi_e,\theta_g}\mathcal{L}_{\text{InfoVDANN}}(\theta_c,\hat{\theta}_d,\phi_e,\theta_g), \tag{13}$$

where symbols with a hat (e.g., $\hat{\theta}_c$) on the right-hand side of (12) and (13) mean that they are fixed when optimizing the target parameters. After training, we extract the transformed features from the $\mathbf{z}$ nodes in Fig. 1.

C. MMD–VDANN and AAE–VDANN

To compute the term $\mathrm{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ in $\mathcal{L}_{\text{InfoVAE}}$, we first need to obtain samples $\mathbf{z}$'s from $q_\phi(\mathbf{z})$. This can be easily addressed by ancestral sampling [32], [35], [44], that is, we first uniformly sample $\mathbf{x}$'s from the training data, and then draw samples $\mathbf{z}$'s from $q_\phi(\mathbf{z}|\mathbf{x})$. Take a mini-batch of B training samples as an example, this can be denoted as follows:

$$\mathbf{z}\sim q_\phi\left(\mathbf{z}|\mathbf{x}_b\right), \quad b\sim \text{ Uniform }(1,\ldots,N). \tag{14}$$

After getting samples from $q_\phi(\mathbf{z}|\mathbf{x}_b)$, the aggregate posterior can be calculated through the Monte Carlo estimate:

$$q_\phi(\mathbf{z}_s)\approx\frac{1}{B}\sum_{b=1}^{B}q_\phi\left(\mathbf{z}_s|\mathbf{x}_b\right), s=1,\ldots,B. \tag{15}$$

If we use these B $\mathbf{z}_s$'s to estimate $\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log q_\phi(\mathbf{z})\right]$, i.e., the first term of $\mathrm{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ in (7), we have

$$\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log q_\phi(\mathbf{z})\right]\approx\frac{1}{B}\sum_{s=1}^{B}\left[\log\frac{1}{B}\sum_{b=1}^{B}q_\phi\left(\mathbf{z}_s|\mathbf{x}_b\right)\right]. \tag{16}$$

Since the estimate in (16) is biased and can only give a lower bound on the true expectation [35], the estimate of $\mathrm{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ will also be biased. This means that we need to set the mini-batch size B to a large value during training to make the estimate of the KL divergence reliable, which would lead to heavy computation.

Although there are other methods to estimate $q_\phi(\mathbf{z})$ and $\mathbb{E}_{q_\phi(\mathbf{z})}\left[\log q_\phi(\mathbf{z})\right]$ [53], [54], these methods are restricted to using KL divergence as the "distance" between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. Inspired by the work in [36], [55], we generalize the KL divergence to other probability distance metrics. Setting $L=1$, the resulting objective of the InfoVAE is then updated as

$$\begin{aligned}\hat{\mathcal{L}}_{\text{InfoVAE}}\left(\theta_g,\phi_e\right) = &-\frac{1}{N}\sum_{r=1}^{R}\sum_{i=1}^{N_r}\Bigg\{\log p_\theta\left(\mathbf{x}_i^r|\mathbf{z}_i^r\right)\\ &-\frac{1-\eta}{2}\sum_{j=1}^{J}\left[\left(\mu_{ij}^r\right)^2+\left(\sigma_{ij}^r\right)^2-1-\log\left(\sigma_{ij}^r\right)^2\right]\Bigg\}\\ &+(\lambda-1+\eta)\mathrm{D}_g\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right),\end{aligned} \tag{17}$$

where $\mathrm{D}_g(\cdot\|\cdot)$ denotes a generalized distance metric.

If we apply MMD [14], [17] as the specialized distance metric in (17), the resulting InfoVDANN is called MMD–VDANN. MMD characterizes the distance between two distributions as the Euclidean distance in the Hilbert space, which can be efficiently computed by the kernel trick. Given a suitable kernel, MMD can match up to infinite moments of their distributions. An unbiased empirical estimate of MMD between datasets $\mathcal{X}$ and $\mathcal{Y}$ is given by

$$\begin{aligned}\mathrm{MMD}^2(\mathcal{X},\mathcal{Y}) = &\frac{1}{N(N-1)}\sum_{n=1}^{N}\sum_{n'\neq n}^{N}k\left(\mathbf{x}_n,\mathbf{x}_{n'}\right)\\ &+\frac{1}{N'(N'-1)}\sum_{n=1}^{N'}\sum_{n'\neq n}^{N'}k\left(\mathbf{y}_n,\mathbf{y}_{n'}\right)\\ &-\frac{2}{NN'}\sum_{n=1}^{N}\sum_{n'=1}^{N'}k\left(\mathbf{x}_n,\mathbf{y}_{n'}\right),\end{aligned} \tag{18}$$

where N and $N^{'}$ denote the number of samples in $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $k\left(\cdot,\cdot\right)$ represents a kernel.

Alternatively, we may use adversarial learning to minimize the distance between two distributions in the latent space as in AAEs [49]. This can be fulfilled by introducing a discriminator to distinguish the samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. We call the InfoVDANN that implements the minimization of $\mathrm{D}_g\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ by adversarial learning as AAE–VDANN.

The optimization of MMD–VDANN and AAE–VDANN is the same as in (11), except that (6) is replaced by (17) for the $\mathcal{L}_{\text{InfoVDANN}}$. Take the MMD–VDANN as an example, we use MMD between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ as the $\mathrm{D}_g\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ term in (17). It is straightforward to implement this by replacing $\mathbf{x}$'s and $\mathbf{y}$'s in (18) with the samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$, respectively. Also, we set both N and $N^{'}$ to the mini-batch size B during training. The latent samples $\mathbf{z}$'s from $q_\phi(\mathbf{z})$ can be drawn according to (14).

IV. Experimental Setup

The performance of various DA methods was evaluated on SRE16 and SRE18-CMN2. All experiments were based on x-vectors [56] using the x-vector extractor available in the Kaldi repository.[2] Unless otherwise stated, the InfoVDANN mentioned in the latter sections represents both MMD–VDANN and AAE–VDANN.

[2] http://kaldi-asr.org/models/m3

A. Training of InfoVDANN, VDANNn and DANN

We used data from four domains as shown in Table I to train the InfoVDANN, VDANN and DANN. Each dataset corresponds to a domain. To briefly summarize, SRE04–10 mainly consist of clean conversational telephone speech in English. Voxceleb1 was a wideband corpus extracted from the YouTube videos spanning a wide range of ethnicities with real-world noises. SwitchBoard-2 was an English corpus of two-sided telephone conversations but was collected in the 90's. Similar to the Voxceleb1 dataset, SITW was a collection from the open source media with unconstrained acoustic conditions. In contrast, the SRE16 evaluation set is composed of telephone conversations spoken in Tagalog and Cantonese, while SRE18-CMN2 contains mainly conversational telephone speech in Tunisian Arabic. Although there is overlap in the collection conditions amongst these datasets, basically, they differed from each other in channels, languages, noises, etc. Therefore, there are mismatches between the training data and the test data and mismatches within the training data.

The statistics of the four training sets are shown in Table I. Note that each training set is a subset of the original set. For example, the minimum number of x-vectors per speaker is 30 for both SRE04–10 and Voxceleb1. SwitchBoard-2 was selected from Phases I–III to ensure that there are at least 20 x-vectors for each speaker, whereas each speaker in SITW has at least 15 x-vectors.

TABLE I
STATISTICS OF TRAINING SETS

Dataset	No. of speakers	No. of utterances
SRE04–10	1,796	53,880
Voxceleb1	1,181	35,430
SwitchBoard-2	268	6,812
SITW	198	3,572

As shown in Fig. 1, there are four sub-networks in the InfoVDANN. The encoder has two hidden layers and each layer has 1,024 nodes. We used ReLU as the activation function in each layer, followed by batch normalization (BN) and dropout. The dimension of the latent space was set to 400. There is only one hidden layer with 2,048 nodes in the decoder. The output layers of both the encoder and decoder are linear. For the speaker classifier, we used a 1024-1024 hidden-layer structure with Leaky ReLU activation functions, and BN and dropout layers were appended after each layer. The output layer has 3,443 nodes with a softmax function, which correspond to 3,443 speakers. The configuration of the domain classifier is similar to that of the speaker classifier except that the number of nodes in the two hidden layers are 128 and 32, respectively. There are four output nodes which correspond to the four domains in Table I. The dropout rate was set to 0.2 for all dropout layers in the network. For the AAE–VDANN, we included an additional latent-variable discriminator to Fig. 1 to differentiate the samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. This discriminator has a 128-16 layered structure followed by ReLU activation and BN in each layer.

We used the Adam optimizer to train the InfoVDANN, VDANN and DANN with a learning rate of 1.0×10^{-3}. The mini-batch size was set to 128. MMD was computed using a mixture of seven radial basis functions (RBFs) with width being set to 0.1, 0.2, 0.4, 1.0, 4.0, 16.0, 256.0, respectively for the MMD–VDANN. For the DANN, we set $\alpha = 0.1$ and $\beta = 0$ in (8), whereas for the VDANN, we set $\alpha = 0.1$ and $\beta = 0.1$ with $\eta = 0$ and $\lambda = 1.0$ in (17). For the InfoVDANN, we set $\alpha = 0.1$, $\beta = 1.0$, $\eta = 0.2$, and $\lambda = 1.0$.

The configuration of the decoder in the VDANN is different from that in [48]. Specifically, the decoder in [48] was implemented by a 1024-1024 hidden-layer structure. Besides, all the networks in this paper were implemented in Tensorflow while those in [48] were based on Keras. Due to these differences in the implementation details, the results of the VDANN and DANN as shown in Table II are different from those in [48].

B. PLDA Training and Scoring

We used the Gaussian PLDA (G-PLDA) backend and the heavy-tailed PLDA (HT-PLDA) [26] for scoring. For SRE16, the baseline G-PLDA and HT-PLDA models were both trained on the augmented SRE04–10 x-vectors. For SRE18, Mixer6 and its augmentation were also added to the training sets. The augmentation step followed the Kaldi's SRE16 recipe. Before G-PLDA training, the x-vectors were centered and projected to a 150 dimensional space by an LDA transformation matrix, followed by whitening and length normalization. The LDA projection matrix was trained on the same dataset as for training the PLDA models. The dimension of the LDA projection was selected according to the EER on the development set. Evidences of why the projection dimension was set to 150 can be found in the supplementary material. As for the HT-PLDA training, we used the same setup as that in [26]. For example, the degree of freedom in the heavy-tailed distribution and the dimension of the speaker subspace were set to 2 and 150, respectively. Also, the LDA projection and length normalization were excluded from the preprocessing.

For all the G-PLDA backends, we also applied Kaldi's PLDA adaptation as an *extra* adaptation step. Specifically, SRE16 unlabeled data were used to adapt the PLDA model for SRE16, whereas we used SRE18 unlabeled data for PLDA adaptation for SRE18. While for the HT-PLDA systems, since Kaldi's PLDA adaptation was not compatible with the HT-PLDA model, we used the unsupervised domain adaptation [5] via parameter interpolation as in [26]. The interpolation factor was set to 0.9 for the out-of-domain part. The adaptation data are the same as those in the G-PLDA adaptation.

For the evaluations of InfoVDANN, VDANN and DANN, we applied the same processing as the baseline using the transformed x-vectors rather than the raw x-vectors.

V. RESULTS AND DISCUSSIONS

A. SRE Performance

We followed the Kaldi's SRE16 recipe for SRE16/18 evaluations for the G-PLDA backends. For the baseline, the x-vectors were centered, LDA-transformed, whitened, and length-normalized before PLDA scoring. The same preprocessing was applied to the transformed x-vectors for the InfoVDANN, VDANN and DANN. As for the preprocessing

TABLE II
PERFORMANCE ON SRE16 AND SRE18-CMN2. THE FIRST PART (ROWS 1–5) AND THE SECOND PART (ROWS 6–10) SHOW THE PERFORMANCE BASED ON THE HEAVY-TAILED PLDA (HT-PLDA) AND THE GAUSSIAN PLDA (G-PLDA) BACKENDS, RESPECTIVELY. THE THIRD PART (ROWS 11–14) PRESENTS THE PERFORMANCE OF SOME STATE-OF-THE-ART END-TO-END DOMAIN ADAPTATION SYSTEMS.

	SRE16-All				SRE16-Cantonese				SRE16-Tagalog				SRE18-CMN2			
	w/o adp		w/ adp		w/o adp		w/ adp		w/o adp		w/ adp		w/o adp		w/ adp	
	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER	minDCF
Baseline (HT-PLDA)	11.48	0.830	12.43	0.871	7.03	0.648	6.43	0.605	**16.36**	0.926	16.84	0.929	**10.42**	0.674	10.21	0.668
DANN (HT-PLDA)	12.03	0.844	12.23	0.850	6.81	0.576	6.37	0.579	16.97	0.940	17.06	0.946	11.06	0.683	10.72	0.665
VDANN (HT-PLDA)	11.57	**0.776**	11.68	**0.800**	6.48	0.568	6.15	0.555	16.51	**0.888**	16.63	**0.897**	10.81	0.674	10.41	0.658
MMD–VDANN (HT-PLDA)	**11.47**	0.797	11.65	0.816	6.24	0.544	6.06	**0.533**	16.40	0.910	16.63	0.905	10.69	**0.655**	10.19	**0.636**
AAE–VDANN (HT-PLDA)	11.50	0.795	**11.62**	0.820	**6.20**	**0.543**	**6.04**	0.539	16.42	0.907	**16.55**	0.917	10.59	0.657	**10.15**	0.643
Baseline (G-PLDA)	11.30	0.890	8.27	0.604	7.16	0.579	4.69	0.427	15.60	0.972	11.93	0.751	11.21	0.676	9.60	0.575
DANN (G-PLDA)	11.64	0.894	8.22	0.604	6.85	0.570	4.33	0.439	16.54	0.973	12.24	0.742	10.77	0.674	9.29	0.574
VDANN (G-PLDA)	10.92	0.852	8.13	0.587	6.62	0.540	4.54	0.413	15.33	0.963	11.76	0.731	10.29	0.664	9.18	0.572
MMD–VDANN (G-PLDA)	10.67	0.835	**7.91**	**0.581**	6.41	0.538	4.07	0.402	**15.03**	0.958	**11.59**	**0.723**	9.92	**0.653**	8.95	0.570
AAE–VDANN (G-PLDA)	**10.61**	**0.832**	**7.91**	0.582	**6.35**	**0.535**	**4.06**	**0.400**	15.08	**0.955**	**11.59**	0.726	**9.91**	0.654	**8.85**	**0.559**
WGAN [23]	13.25	0.899	9.15	0.677	7.39	0.561	–	–	19.12	0.968	–	–	10.35	0.658	9.21	0.602
WGAN+lan+sup [23]	9.59	0.746	**8.00**	0.651	**5.59**	**0.497**	–	–	**13.70**	**0.880**	–	–	8.88	0.619	8.25	0.576
LSGAN [24]	11.74	–	–	–	7.90	–	–	–	15.63	–	–	–	–	–	–	–
Multi-level [16]	**9.03**	**0.585**	8.29	**0.546**	–	–	–	–	–	–	–	–	**8.33**	**0.520**	**8.09**	**0.521**

of x-vectors for HT-PLDA models, only centering and whitening were applied.

Table II shows the performance of different systems on SRE16 and SRE18-CMN2. The first part (rows 1–5) and the second part (rows 6–10) are the results applying HT-PLDA and G-PLDA scoring, respectively. For evaluations based on HT-PLDA, without PLDA adaptation, the HT-PLDA backends only outperform the G-PLDA counterparts in minDCF under SRE16-All and SRE16-Tagalog. For SRE18-CMN2 evaluations, the HT-PLDA backends even cannot compete with the G-PLDA models. Besides, we see that the unsupervised DA failed on SRE16-All and SRE16-Tagalog, and it only worked on SRE16-Cantonese and SRE18-CMN2 with a slight performance gain over the unadapted version. In general, there is no consistent observation that InfoVDANN with HT-PLDA backends outperforms the other systems. It seems that although the HT-PLDA model is theoretically capable of addressing the non-Gaussian embeddings, it failed to achieve consistent gains compared with those of the G-PLDA backend (rows 6–10). Maybe the (transformed) x-vectors do not present a strong heavy-tailed characteristic. Or there may be a need for LDA to find more discriminative directions in the speaker subspace before performing HT-PLDA scoring.

The performance based on the G-PLDA backend is shown in the second part (rows 6–10) in Table II. Without Kaldi's PLDA adaptation under SRE16-All, we can observe that although VDANN can reduce domain mismatch in terms of both EER and minDCF, both MMD–VDANN and AAE–VDANN consistently outperform the VDANN. We also performed Kaldi's PLDA adaptation as an *extra* domain adaptation. From columns 4–5 in the second part, we see that Kaldi's PLDA adaptation is still helpful in further reducing domain mismatch. From the improvement due to the PLDA adaptation, we may conclude that adversarial domain adaptation and PLDA adaptation are complementary. The fact that the performance of InfoVDANN is better than that of DANN verifies that imposing the variational regularization on the transformed x-vectors is effective for domain adaptation. The performance of the Cantonese and Tagalog partitions is consistent with that of SRE16-All. These findings suggest both types of InfoVDANNs (MMD–VDANN and AAE–VDANN) benefit DA in extracting additional speaker discriminative information from the training data compared with the VDANN, and that incorporating an MI term in the loss function is effective for reducing domain mismatch.

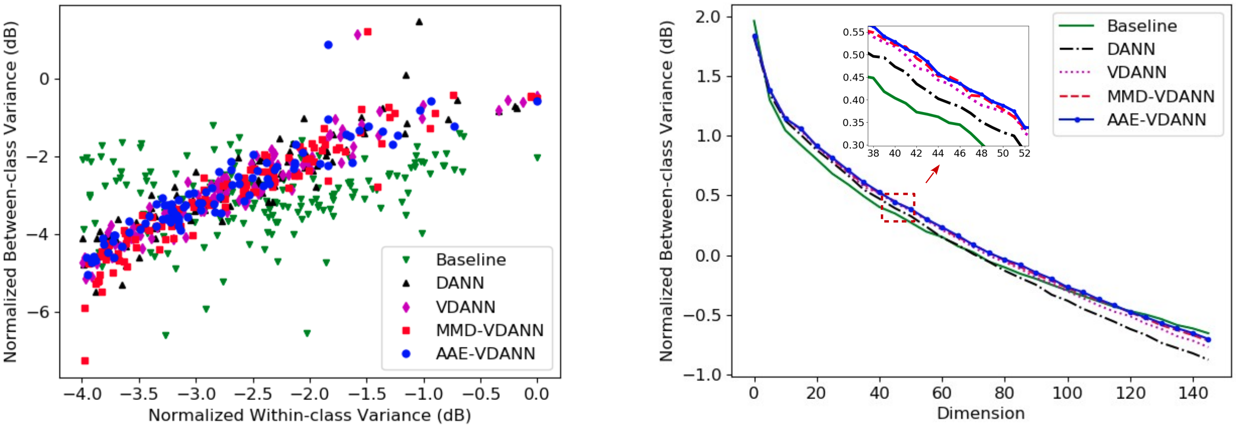
From the last four columns in the second part of Table II, we obtain similar conclusions for SRE18-CMN2 as in SRE16: maintaining high MI between the latent features and the inputs can feed more speaker information into the learned embeddings, which enhances speaker recognition performance.

As shown in the supplementary material, the P-values of the McNemar's tests [57] between both InfoVDANNs and the others are mostly zeros for SRE16-All and SRE18-CMN2. This means that the improvement of both InfoVDANNs over VDANN, DANN and the baseline is statistically significant.

We also present the performance of some state-of-the-art end-to-end systems in the third part (rows 11–14) of Table II. Generally, results show that end-to-end systems outperform the embedding–backend cascaded systems. This is reasonable because the end-to-end systems have greater capacity to learn the domain invariance.

B. Speaker Discriminative Features

By explicitly incorporating an MI term in the objective function, InfoVDANN is able to feed *extra* information into the speaker embeddings, making them more discriminative than those of the VDANN and DANN. To investigate the discriminativeness of the InfoVDANN, we plot the between-class variances against the within-class variances using the SRE04–10 dataset (with augmentation). In Fig. 2(a), the x-axis denotes the logarithm of normalized within-class variances: $\log[(V_w - \min(V_w))/(\max(V_w) - \min(V_w))]$, where V_w is



(a)

(b)

Fig. 2. Illustration for (a) between-class variances versus within-class variances before LDA and (b) between-class variances in each dimension after LDA.

the within-class variance. Therefore, different systems can be compared in the same scale. The scale of the y-axis is $\log[(V_b - \min(V_w))/(\max(V_w) - \min(V_w))]$, where V_b denotes the between-class variance. We see that InfoVDANN generally has larger between-class variances than the baseline for each within-class variance. This means that the x-vectors transformed by the InfoVDANN are more discriminative compared with the baseline because the speaker clusters become more separated after the transformation. However, it is not clear if InfoVDANN outperforms the VDANN and DANN because their scatters overlap with each other in the plot.

Since LDA was applied before PLDA scoring, we further evaluate the discriminativeness of these systems after LDA projection. The sorted between-class variances after normalization were plotted in Fig. 2(b). Note that the within-class variances are all normalized to 1. According to Fig. 2(b), we see that for the first 150 dimensions, InfoVDANN consistently has larger between-class variances than the other systems. This suggests that the embeddings transformed by InfoVDANN are more discriminative after LDA projection, which verifies the advantage of InfoVDANN over the VDANN and DANN.

C. Effect on Gaussian Regularization

The InfoVDANN and the VDANN apply variational regularization on the learned embeddings so that the transformed x-vectors will follow a Gaussian distribution. To investigate the effectiveness of this Gaussian regularization, we present the normal Q–Q plots [58] of two randomly selected dimensions of the raw x-vectors and the x-vectors transformed by InfoVDANN, VDANN, and DANN. These x-vectors were selected from the CMN2 part of the SRE18 evaluation set. Evidently, as shown in Fig. 3, the distributions of all the x-vectors transformed by the MMD–VDANN, AAE–VDANN, and VDANN are closer to a Gaussian distribution than the DANN and the baseline systems, whereas the InfoVDANN-transformed ones seem to be more Gaussian than those transformed by the VDANN. This suggests that the InfoVAE loss can make the latent vectors $\mathbf{z}$'s to follow a Gaussian distribution. The p-values obtained from Shapiro-Wilk tests [59], [60] also suggest that the distributions of InfoVDANN-transformed x-vectors are closer to the standard Gaussian than the x-vectors transformed by the other three methods.

D. Comparison of Mutual Information

The InfoVDANN explicitly incorporates an MI term in the objective function during DAT to additionally preserve meaningful information between the learned features and the input set. It makes sense to infer latent representations that are more speaker discriminative by maximizing this MI term together with the optimization of other sub-networks in the InfoVDANN. It has been verified in Table II that both the InfoVDANNs consistently outperform the VDANN based on the G-PLDA backend. To further evaluate the effectiveness of this MI maximization, we report the MI estimates on both the SRE16 Evaluation set and SRE18-CMN2 Evaluation set in Table III.

The MI between the latent variable $\mathbf{z}$ and the input $\mathbf{x}$ is estimated by (4) in the supplementary material. We randomly selected 1,024 x-vectors from each set (e.g., SRE16-eval Enrollment and Test, SRE18-eval-CMN2 Enrollment and Test), and used these 1,024 samples as one single batch for each estimation. Each of the mean and variance was based on 200 simulations.

We can see from Table III that both MMD–VDANN and AAE–VDANN have higher MI between the learned features and the inputs than the VDANN, which contributes to the performance gain in SRE16 and SRE18-CMN2. According to (5) in the supplementary material, the MI estimates are bounded by 6.9314 (i.e., $\log(1024)$) for these samples. Although there is some gap between the estimates and the upper bound, the performance gains on SREs are still statistically significant as reported in Subsection V-A.

E. Impact of Hyperparameters λ and η

In (8), we use four hyperparameters (α, β, λ and η) in the InfoVDANN's objective function. VDANN is a special case

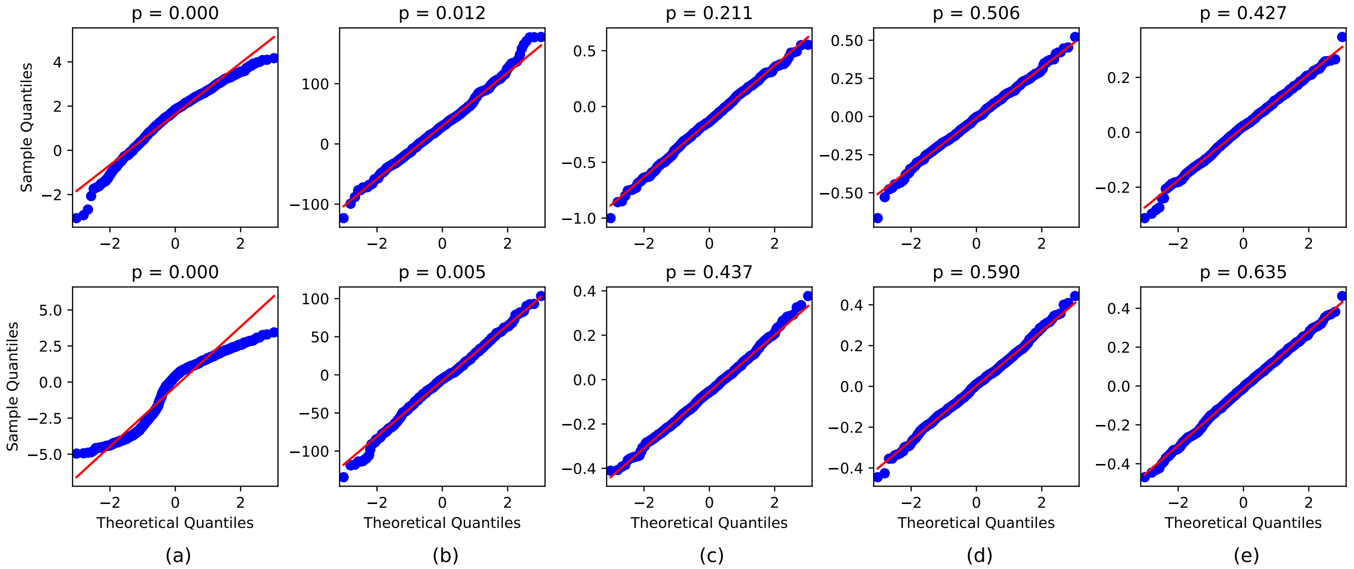


Fig. 3. Quantile-quantile (Q–Q) plots of the 151-st (Row 1), and 301-st (Row 2) components of (a) raw x-vectors, (b) DANN-transformed x-vectors, (c) VDANN-transformed x-vectors, (d) MMD–VDANN transformed x-vectors, and (e) AAE–VDANN transformed x-vectors. The vertical and horizontal axes correspond to the samples under test and the samples drawn from a standard normal distribution, respectively. The red line represents the case of perfectly Gaussian. The p-values above the graphs were obtained from Shapiro-Wilk tests, with $p > 0.05$ meaning failing to reject the null hypothesis that the test samples come from a Gaussian distribution (i.e., the larger the p, the more Gaussian the distribution).

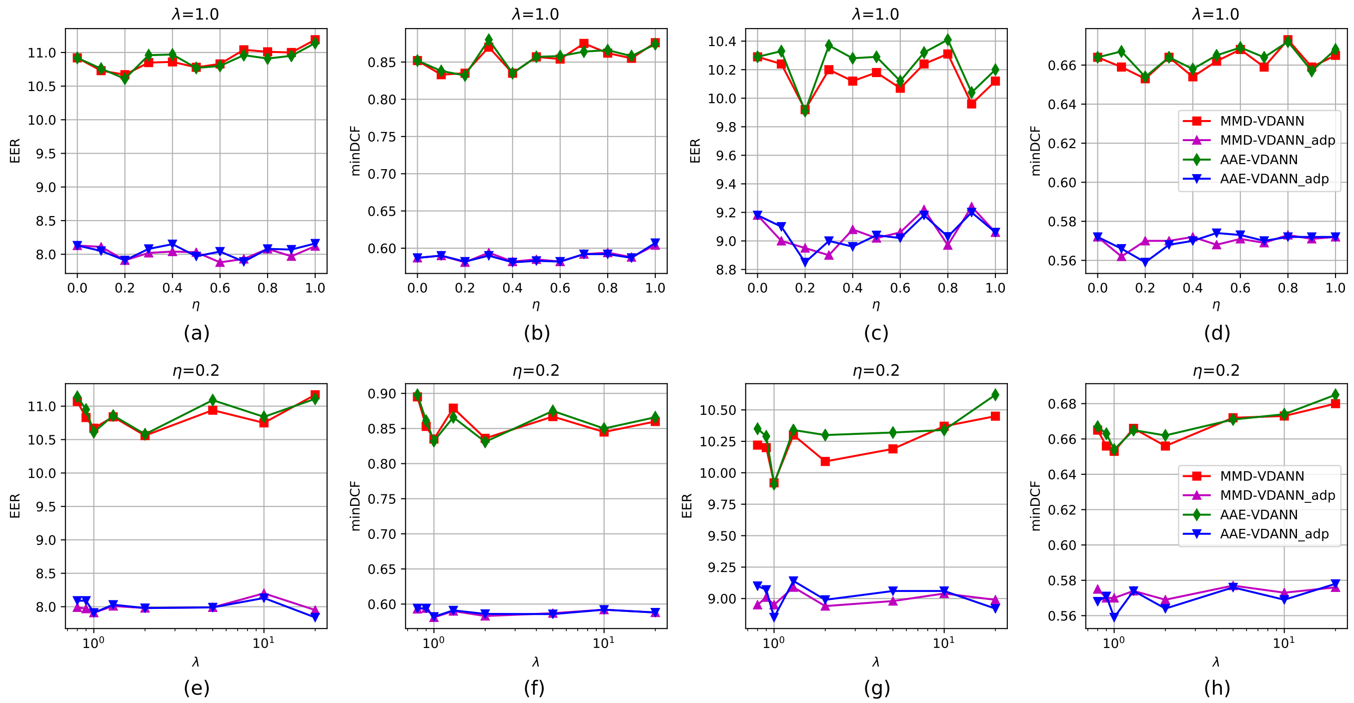
TABLE III
ESTIMATES OF THE MUTUAL INFORMATION TERM ($I_q(\mathbf{x};\mathbf{z})$ IN (4)) UNDER SRE16 EVALUATION SET AND SRE18-CMN2 EVALUATION SET

	SRE16-eval				SRE18-eval-CMN2			
	Enrollment		Test		Enrollment		Test	
	mean	variance	mean	variance	mean	variance	mean	variance
VDANN	4.466	1.092	5.078	1.115	3.922	1.045	4.567	1.077
MMD–VDANN	4.811	1.052	5.770	1.150	5.357	1.228	5.028	1.327
AAE–VDANN	5.114	1.047	6.263	1.151	5.038	1.163	5.031	1.248

of InfoVDANN in that the InfoVDANN loss degenerates into the VDANN objective when $\eta = 0$, and $\lambda = 1$ in (17). The discrepancy between the two loss functions is the choice of $\mathrm{D}_g(q_\phi(\mathbf{z})\|p(\mathbf{z}))$ and the contribution of $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$ and $\mathrm{D}_g(q_\phi(\mathbf{z})\|p(\mathbf{z}))$ to the total loss, which are controlled by λ and η, respectively. To examine the superiority of InfoVDANN over VDANN, we present the impact of λ and η on SRE performance in Fig. 4 by setting $\alpha = 0.1$, and $\beta = 1.0$ in (8).

From (5), we note that λ is to balance the variational inference and data reconstruction during the optimization of the VAE sub-network in Fig. 1. In our experimental setup, because the difference between the dimension of the latent vector (400) and that of the input embedding (512) is not very large, we started with $\lambda = 1.0$ to evaluate the influence of η on SRE16-eval and SRE18-CMN2. As shown in Fig. 4(a) and Fig. 4(b), both MMD–VDANN and AAE–VDANN achieve the best performance at $\eta = 0.2$ for SRE16 with and without the *extra* Kaldi's PLDA adaptation. In this regard, we fixed η to 0.2 when inspecting how λ impacts SRE16 performance. The result is illustrated in Fig. 4(e) and Fig. 4(f), from which we can see that within a wide range of $\lambda \in [0.8, 10]$, the SRE16 performance without PLDA adaptation does not change too much, while it achieves a slightly better performance at $\lambda = 1.0$. With Kaldi's PLDA adaptation, we can obtain similar performance at $\lambda = 1.0$ and $\lambda = 2.0$. Overall, with $\eta = 0.2$, both MMD–VDANN and AAE–VDANN obtained a consistently good performance at $\lambda = 1.0$ for SRE16 with and without PLDA adaptation. From Figs. 4(a), 4(b), 4(e) and 4(f), we conclude that $\eta = 0.2$ and $\lambda = 1.0$ are suitable choices for compromising the contribution of $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$ and $\mathrm{D}_g(q_\phi(\mathbf{z})\|p(\mathbf{z}))$ to produce meaningful speaker embeddings.

We obtained a similar conclusion for SRE18-CMN2, as shown in Figs. 4(c), 4(d), 4(g) and 4(h). The performance over the choice of both hyperparameters with PLDA adaptation is more stable than that without it. For fixed λ, increasing η from 0 to an appropriate value (e.g., 0.2) can improve the performance of InfoVDANN. However, a larger value for η can have a detrimental effect on the performance. Because η represents the contribution of the MI term in (5), maintaining a high MI between the latent features and the inputs during training does not always produce better features. Note that when $\eta = 1.0$ and $\lambda = 1.0$, (5) becomes the standard AAE loss function [49]. This in turn verifies that $\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$ is still very important as a regularization term on the learned embeddings. On the other hand, with $\eta = 0.2$, as shown in



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Impact of hyperparameters λ and η on the performance of SRE16 and SRE18-CMN2. The first row shows the impact of η on the performance of SRE16 [(a) and (b)] and SRE18-CMN2 [(c) and (d)] with λ fixed to 1.0. The second row illustrates the impact of λ on SRE16 [(e) and (f)] and SRE18-CMN2 [(g) and (h)] with $\eta = 0.2$. The instances with a suffix "_adp" in the legend denote the use of Kaldi's PLDA adaptation.

Figs. 4(g) and 4(h), the performance is insensitive to a wide range of λ.

From the above analysis, we thus used $\alpha = 0.1$, $\beta = 1.0$, $\eta = 0.2$, and $\lambda = 1.0$ for experimental comparisons in the previous subsections. From Figs. 4(a)–(h), we also observe that there is no big difference between the choice of $\mathrm{D}_g\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$, e.g., MMD and adversarial learning, since MMD–VDANN has a comparable performance with the AAE–VDANN for nearly all of the experiments. This also empirically verifies the robustness of InfoVDANN using different ways to minimize the generalized divergence. But due to the additional latent-variable discriminator (distinguishing the samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$) in AAE–VDANN, AAE–VDANN possesses more parameters compared with MMD–VDANN. Moreover, the minimax optimization of this latent-variable discriminator is not as efficient as the minimization of the MMD. As such, in practice, training AAE–VDANN takes slightly longer than training MMD–VDANN. Thus, from the perspective of implementation, MMD–VDANN may be a better choice.

VI. Conclusions

In this paper, we proposed a network called InfoVDANN for unsupervised domain adaptation (DA). The InfoVDANN incorporated an InfoVAE into the DANN to encourage higher mutual information (MI) between the learned features and the inputs, while simultaneously retaining the advantage of VDANN as a Gaussian distribution regularizer. Experimental results on SRE16 and SRE18-CMN2 showed that the InfoVDANN is capable of reducing domain mismatch through domain adversarial training and maintaining high speaker information in the transformed features. Gaussianity tests verified the effectiveness of the variational regularization. The fact that the InfoVDANN consistently outperforms VDANN suggests that feeding suitable MI into the training of InfoVDANNs is effective for extracting *extra* information for speaker verification. The consistency of the MI estimates on the test datasets also confirmed the feasibility of using InfoVDANNs for unsupervised DA.

References

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

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

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

[4] S. Shum, D. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 266–272.

[5] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 378–383.

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

[7] 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.

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

[9] M. Rahman, A. Kanagasundaram, D. Dean, and S. Sridharan, "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proc. Annual Conference of the International Speech Communication Association*, 2015, pp. 1017–1021.

[10] J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 176–180.

[11] P. Bousquet and M. Rouvier, "On robustness of unsupervised domain adaptation for speaker recognition," in *Proc. Annual Conference of the International Speech Communication Association*, 2019, pp. 2958–2962.

[12] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 5821–5825.

[13] S. Shon, S. Mun, W. Kim, and H. Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," in *Proc. Annual Conference of the International Speech Communication Association*, 2017, pp. 1014–1018.

[14] W. Lin, M. Mak, and J. Chien, "Multi-source i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 16, no. 12, pp. 2412–2422, 2018.

[15] W. Lin, M. Mak, Y. Tu, and J. Chien, "Semi-supervised nuisance-attribute networks for domain adaptation," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6236–6240.

[16] W. Lin, M. W. Mak, N. Li, D. Su, and D. Yu, "Multi-level deep neural network adaptation for speaker verification using MMD and consistency regularization," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2020, pp. 6839–6843.

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

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2130, 2016.

[20] J. Tsai and J. Chien, "Adversarial domain separation and adaptation," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, Sep. 2017, pp. 1–6.

[21] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B. Juang, "Speaker-invariant training via adversarial learning," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[22] Q. Wang, W. Rao, S. Sun, L. Xie, E. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2018, pp. 4889–4893.

[23] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6006–6010.

[24] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6226–6230.

[25] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2010.

[26] A. Silnova, N. Brümmer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast variational Bayes for heavy-tailed PLDA applied to i-vectors and x-vectors," in *Proc. Annual Conference of the International Speech Communication Association*, 2018, pp. 72–76.

[27] D. Garcia-Romero and C. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Annual Conference of the International Speech Communication Association*, 2011, pp. 249–252.

[28] L. Li, Z. Tang, Y. Shi, and D. Wang, "Gaussian-constrained training for speaker verification," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6036–6040.

[29] J. Chien and K. Peng, "Adversarial learning and augmentation for speaker recognition," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 342–348.

[30] D. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations*, 2014.

[31] Y. Zhang, L. Li, and D. Wang, "VAE-based regularization for deep speaker embedding," in *Proc. Annual Conference of the International Speech Communication Association*, 2019, pp. 4020–4024.

[32] M. Hoffman and M. Johnson, "ELBO surgery: yet another way to carve up the variational evidence lower bound," in *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2016.

[33] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. Saurous, and K. Murphy, "Fixing a broken ELBO," in *Proc. International Conference on Machine Learning*, 2018, pp. 159–168.

[34] R. Shu, H. Bui, S. Zhao, M. Kochenderfer, and S. Ermon, "Amortized inference regularization," in *Advances in Neural Information Processing Systems*, 2018, pp. 4393–4402.

[35] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in *International Conference on Learning Representations*, 2019.

[36] S. Zhao, J. Song, and S. Ermon, "InfoVAE: balancing learning and inference in variational autoencoders," in *Proc. AAAI Conference on Artificial Intelligence*, 2019, pp. 5885–5892.

[37] J. Tomczak and M. Welling, "VAE with a VampPrior," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1214–1223.

[38] C. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-VAE," in *arXiv preprint arXiv:1804.03599*, 2018.

[39] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[40] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. International Conference on Machine Learning*, 2015, pp. 1530–1538.

[41] D. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Amortized inference regularization," in *Advances in Neural Information Processing Systems*, 2016, pp. 4743–4751.

[42] S. Zhao, J. Song, and S. Ermon, "Towards deeper understanding of variational autoencoding models," in *arXiv preprint arXiv:1702.08658*, 2017.

[43] Y. Kim, S. Wiseman, A. Miller, D. Sontag, and A. Rush, "Semi-amortized variational autoencoders," in *Proc. International Conference on Machine Learning*, 2018, pp. 2678–2687.

[44] A. Dieng, Y. Kim, A. Rush, and D. Blei, "Avoiding latent variable collapse with generative skip models," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2397–2405.

[45] J. Chien and C. Wang, "Self attention in variational sequential learning for summarization," in *Proc. Annual Conference of the International Speech Communication Association*, 2019, pp. 1318–1322.

[46] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[47] J. Ho, X. Chen, A. Srinivas, R. Duan, and P. Abbeel, "Flow++: Improving flow-based generative models with variational dequantization and architecture design," in *Proc. International Conference on Machine Learning*, 2019, pp. 2722–2730.

[48] Y. Tu, M. Mak, and J. Chien, "Variational domain adversarial learning for speaker verification," in *Proc. Annual Conference of the International Speech Communication Association*, 2019, pp. 4315–4319.

[49] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," in *arXiv preprint arXiv:1511.05644*, 2015.

[50] J. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media, 2008.

[51] M. Mak and J. Chien, *Machine Learning for Speaker Recognition*. Cambridge University Press, 2020.

[52] M. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, and Y. Bengio, "Mutual information neural estimation," in *Proc. International Conference on Machine Learning*, 2018, pp. 531–540.

[53] R. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 2615–2625.

[54] B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. Brooks, J. Dy, and J. Meent, "Structured disentangled representations," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2525–2534.

[55] H. Kim and A. Mnih, "Disentangling by factorising," in *arXiv preprint arXiv:1802.05983*, 2018.

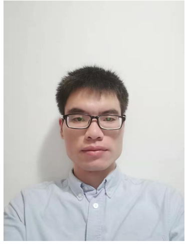
[56] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2018, pp. 5329–5333.

[57] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 532–535.

[58] M. B. Wilk and R. Gnanadesikan, "Probability plotting methods for the analysis of data," *Biometrika*, vol. 55, no. 1, pp. 1–17, 1968.

[59] S. Shapiro and M. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[60] N. M. Razali and Y. B. Wah, "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests," *Journal of Statistical Modeling and Analytics*, vol. 2, no. 1, pp. 21–33, 2011.



Youzhi TU received a B.Eng. degree and an M.Sc. degree from Harbin Engineering University in 2012 and 2015, respectively. Since 2018, 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 and machine 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 a Professor in the same department. He has authored more than 190 technical articles in speaker recognition, machine learning, and bioinformatics. Dr. Mak also coauthored postgraduate textbooks *Biometric Authentication: A Machine Learning Approach*, Prentice-Hall, 2005 and *Machine Learning for Speaker Recognition*, Cambridge University Press, 2020. 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, 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. He served as the associate editor of the IEEE Signal Processing Letters in 2008-2011, the general co-chair of the IEEE International Workshop on Machine Learning for Signal Processing in 2017, and the tutorial speaker of ICASSP, INTERSPEECH, COLING, ACL, AAAI, IJCAI, KDD and ICME. 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.