INFORMATION MAXIMIZED VARIATIONAL DOMAIN ADVERSARIAL LEARNING FOR SPEAKER VERIFICATION

Youzhi Tu and Man-Wai Mak

The Hong Kong Polytechnic University
Dept. of Electronic and Information Engineering, Hong Kong SAR

Jen-Tzung Chien

National Chiao Tung University
Dept. of Electrical and Computer Engineering, Taiwan

ABSTRACT

Domain mismatch is a common problem in speaker verification. This paper proposes an information-maximized variational domain adversarial neural network (InfoVDANN) to reduce domain mismatch by incorporating an InfoVAE into domain adversarial training (DAT). DAT aims to produce speaker discriminative and domain-invariant features. The InfoVAE has two roles. First, it performs variational regularization on the learned features so that they follow a Gaussian distribution, which is essential for the standard PLDA backend. Second, it preserves mutual information between the features and the training set to extract *extra* speaker discriminative information. Experiments on both SRE16 and SRE18-CMN2 show that the InfoVDANN outperforms the recent VDANN, which suggests that increasing the mutual information between the latent features and input features enables the InfoVDANN to extract *extra* speaker information that is otherwise not possible.

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

1. INTRODUCTION

To achieve optimal performance, speaker verification (SV) systems rely on the condition that the training data share the same distribution with the test data. In practice, however, this condition is often not satisfied and domain mismatch occurs, posing a great challenge to SV. Usually, domain adaptation (DA) is adopted to alleviate this problem.

Recent research on DA has been focusing on the unsupervised situation where only some unlabeled target-domain data are available besides large amount of labeled source-domain data. One approach is to hypothesize speaker labels through clustering; with hypothesized labels, one can adapt the probabilistic linear discriminant analysis (PLDA) model to the target domain [1, 2]. Another category aims to learn a domain-invariant space for transforming the source-domain i-vectors [3], e.g., inter-dataset variability compensation [4] and dataset-invariant covariance normalization [5]. In [6, 7], Lin *et al.* applied maximum mean discrepancy (MMD) [8] as a distribution distance metric for training autoencoders and produced features that are more invariant to multiple domains. Since the emergence of generative adversarial networks [9], adversarial learning has been applied for DA to create a domain-invariant space [10, 11]. In [12], Wang *et al.* utilized domain adversarial training (DAT) [10] to generate speaker discriminative and domain-invariant representations, which outperforms traditional DA approaches on Domain Adaptation Challenge 2013. Rohdin *et al.* [13] followed the same framework but 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 performance of SV systems 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 [14, 15] or applying the i-vector length normalization [16]. 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 work trying to Gaussianize speaker embeddings obtained by neural networks. For instance, in [17], Tu *et al.* proposed a variational domain adversarial neural network (VDANN) by incorporating a variational autoencoder (VAE) [18] into the standard DANN [10, 12] to regularize the distribution of embedded features to be Gaussian. The transformed embeddings have been shown to be more Gaussian than the DANN-transformed ones, which led to performance improvement in SRE16 and SRE18-CMN2. A similar approach using VAEs for Gaussian regularization for speaker embeddings was proposed in [19].

Using VAEs to regularize the latent variables has achieved some progress [17, 19]. However, training VAEs by maximizing the evidence lower bound (ELBO) has some problems which can cause failure in learning useful latent representations [20, 21, 22]. One problem is that given finite training

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 152137/17E, and Taiwan MOST, Grant No. 108-2634-F-009-003.

data, a VAE tends to overfit the training set and generate inaccurate variational posteriors [23]. Another problem is that if the decoder is flexible enough, a VAE can produce noninformative latent vectors independent of the input [24, 25, 26]. This is undesirable if our objective is to learn meaningful representations. In [23], Zhao *et al.* proposed an InfoVAE to address these problems. The idea is to increase the contribution of the KL divergence between the aggregated variational posterior [21, 27] and the latent prior so that latent inference and data reconstruction can be balanced. Also, by explicitly adding a mutual information (MI) term to the objective function, the dependence of the latent vectors on the input can be enhanced. In this paper, we adopt the idea of InfoVAE and extend the VDANN [17] for unsupervised DA. With the InfoVAE, the learned features can gain more meaningful information from the input, while simultaneously retain the benefit of VDANN to produce speaker discriminative, domain-invariant and Gaussian distributed features. We call the resulting network InfoVDANN in this paper.

2. INFO VARIATIONAL AUTOENCODERS

Suppose we have a training set $\mathcal{X}$ whose true data distribution is denoted as $p_{\mathcal{D}}(\mathbf{x})$ and the underlying generation is determined by the latent variable set $\mathcal{Z}$. For $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$, a VAE can be optimized by maximizing the ELBO [23]:

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x})}\left[-\operatorname{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right)\right. \\ &\quad \left.+\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]\right], \qquad (1)\\ &\propto -\operatorname{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right) \\ &\quad -\mathbb{E}_{q_\phi(\mathbf{z})}\left[\operatorname{KL}\left(q_\phi(\mathbf{x}|\mathbf{z})\|p_\theta(\mathbf{x}|\mathbf{z})\right)\right], \qquad (2)\end{aligned}$$

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 $\mathbf{z}$. $q_\phi(\mathbf{z})$ in Eq. 2 is the aggregated posterior [21, 27]:

$$q_\phi(\mathbf{z}) = \int_{\mathbf{x}} p_{\mathcal{D}}(\mathbf{x}) q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{x}. \qquad (3)$$

Eq. 3 suggests that $q_\phi(\mathbf{z})$ cannot be computed exactly, because it requires an aggregation over the entire training set. In practice, we can approximate the integral in Eq. 3 by a Monte Carlo estimate [28, 29].

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 posterior even though the ELBO can be maximized to infinity [23]. This limitation is exacerbated when the dimension of the latent variables is much lower than the input dimension, i.e., optimization tends to sacrifice variational inference to enhance data reconstruction. Because from Eq. 2 we find that 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 cause overfitting easily given finite data. Second, if the decoder is flexible enough, VAE training will ignore the information in the latent features related to the input. As a result, the MI between the latent features and the input vanishes, leading to noninformative representations [30, 31, 32]. We also call this issue as the posterior collapse [22, 33, 34]. In this case the learned features will not depend on the training data. This is undesirable as our objective is to learn meaningful representations for unsupervised DA.

Zhao *et al.* [23] proposed a new objective based on Eq. 2 to address the problems in VAEs by 1) adding a scalar to increase the contribution of $\operatorname{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ and counteract the dimension imbalance between $\mathcal{X}$ and $\mathcal{Z}$; 2) incorporating an MI term which explicitly retains high mutual information between $\mathbf{x}$ and $\mathbf{z}$. The resulting model is called the InfoVAE whose objective is expressed as follows:

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

where $I_q(\mathbf{x};\mathbf{z})$ is the MI between $\mathbf{x}$ and $\mathbf{z}$ under $q_\phi(\mathbf{x},\mathbf{z})$. λ compensates for the dimension imbalance between $\mathbf{x}$ and $\mathbf{z}$, so that 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 Eq. 4 as Eq. 5 because the MI term cannot be computed directly. Note that we can further generalize the $\operatorname{KL}\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right)$ in Eq. 5 to broader divergence families for efficient optimization, e.g., we may use MMD [8] as a divergence measure or introduce a discriminator and apply adversarial training to distinguish samples from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ as in the adversarial autoencoder (AAE) [27].

3. INFORMATION MAXIMIZED VARIATIONAL DOMAIN ADVERSARIAL NEURAL NETWORK

Due to the problems in training VAEs described in Section 2, we propose an InfoVDANN by incorporating an InfoVAE into DANN to learn features that can gain more meaningful information from the input while simultaneously leverage the benefit of the VDANN.

As shown in Figure 1, the InfoVDANN has a similar structure as the VDANN. It consists of a speaker predictor C, a domain classifier D and a VAE which contains 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 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 $\mathbf{y}$ as the one-hot speaker labels. For $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, we define

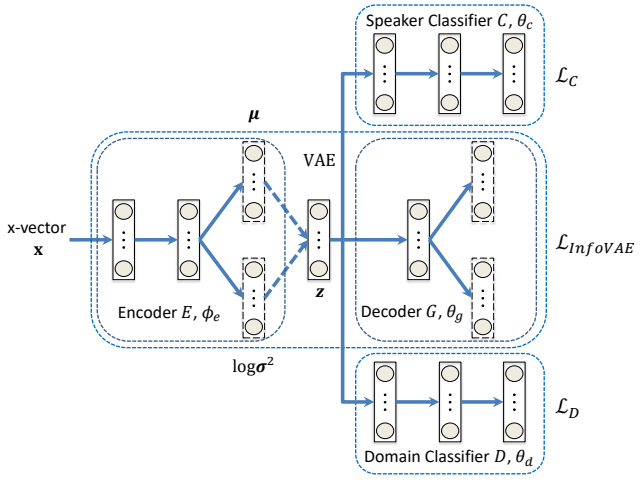

Fig. 1. Schematic of InfoVDANN. The solid and dashed arrows represent network connections and stochastic sampling, respectively. After training, the transformed features are extracted from the $\mathbf{z}$ node.

the InfoVDANN loss as:

$$\mathcal{L}_{\text{InfoVDANN}}(\theta_c, \theta_d, \phi_e, \theta_g) = \mathcal{L}_C(\theta_c, \phi_e) - \alpha\mathcal{L}_D(\theta_d, \phi_e) + \beta\mathcal{L}_{\text{InfoVAE}}(\phi_e, \theta_g), \quad (6)$$

where

$$\mathcal{L}_C(\theta_c, \phi_e) = \sum_{r=1}^{R} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}^{(r)})} \left\{ -\sum_{k=1}^{K} y_k^{(r)} \log C\left(E\left(\mathbf{x}^{(r)}\right)\right)_k \right\}, \quad (7)$$

$$\mathcal{L}_D(\theta_d, \phi_e) = \sum_{r=1}^{R} \mathbb{E}_{p_{\mathcal{D}}(\mathbf{x}^{(r)})} \left\{ -\log D\left(E\left(\mathbf{x}^{(r)}\right)\right)_r \right\}, \quad (8)$$

and

$$\mathcal{L}_{\text{InfoVAE}}(\theta_g, \phi_e) = -\sum_{r=1}^{R}\sum_{i=1}^{N_r} \left\{ \log p_\theta\left(\mathbf{x}_i^{(r)} | \mathbf{z}_i^{(r)}\right) + \frac{1-\eta}{2} \sum_{j=1}^{J} \left[1 + \log\left(\sigma_{ij}^{(r)}\right)^2 - \left(\mu_{ij}^{(r)}\right)^2 - \left(\sigma_{ij}^{(r)}\right)^2 \right] - (\lambda - 1 + \eta) \mathrm{D}_g\left(q_\phi(\mathbf{z}_i^{(r)}) \| p(\mathbf{z}_i^{(r)})\right) \right\}. \quad (9)$$

The subscript k in Eq. 7 indexes the speakers. Eq. 9 is the negative of Eq. 5 where J denotes the dimension of $\mathbf{z}$ and $\mathrm{D}_g(\cdot\|\cdot)$ is a generalized divergence measure which can be implemented by MMD or adversarial training. α and β control the contribution of $\mathcal{L}_C$ and $\mathcal{L}_{\text{InfoVAE}}$, respectively.

During training, for each mini-batch, we first optimize D by minimizing $\mathcal{L}_D(\theta_d, \phi_e)$. θ_d are then fixed while training the remaining parts of the InfoVDANN. The min-max optimization can be summarized as follows:

$$\min_{\theta_c, \phi_e, \theta_g} \max_{\theta_d} \mathcal{L}_{\text{InfoVDANN}}(\theta_c, \theta_d, \phi_e, \theta_g). \quad (10)$$

After training, we may sample the transformed features from the $\mathbf{z}$ node as shown in Figure 1.

4. RELATION TO PRIOR WORK

In [12], DANN was applied to produce speaker discriminative and domain-invariant features. In that case, we have $R = 2$. The DANN is a special case of the InfoVDANN. Specifically, if we remove the decoder and the sampling procedure in the InfoVDANN, we obtain the DANN.

Since there is no extra constraint on the distribution of features learned from a DANN, the adversarial training may lead to non-Gaussian latent vectors, which do not meet the Gaussianity requirement of the PLDA backend. VDANN [17] was proposed to overcome this limitation. By incorporating a VAE into DAT, we are able to regularize the learned features so that they are Gaussian distributed after VDANN transformation. We find that the VDANN also belongs to the InfoVDANN class: by setting $\eta = 0$, and $\lambda = 1$, Eq. 9 becomes the VDANN objective.

5. EXPERIMENTAL SETUP

The performance of various DA methods were evaluated on SRE16 and SRE18-CMN2. The x-vector extractor available from the Kaldi repository[1] was used for extracting x-vectors [35] in the experiments.

5.1. InfoVDANN, VDANN and DANN training

We used data from four domains to train the InfoVDANN, VDANN and DANN. The statistics of the training data are shown in Table 1.

Table 1. Statistics of training sets

Dataset	No. of speakers	No. of utterances
SRE04–10	1,806	54,180
Voxceleb1	1,251	37,530
SwitchBoard II	273	6,962
SITW	203	3,700

As shown in Figure 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). 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 3,533 nodes with a softmax function which correspond to 3,533 speakers. The configuration of the domain classifier is similar to that of the speaker classifier except that the number of

[1] http://kaldi-asr.org/

nodes in the two hidden layers are 128 and 32, respectively. It has four nodes in the output layer corresponding to the four domains in Table 1.

To train the InfoVDANN, we used two divergence metrics to measure the discrepancy between $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$ in Eq. 9: Maximum mean discrepancy (MMD) [6] and adversarial training used in the AAE [27]. The resulting networks are called MMD-VDANN and AAE-VDANN, respectively. The discriminator in the AAE-VDANN is to differentiate the samples drawn from $q_\phi(\mathbf{z})$ and $p(\mathbf{z})$. It has a 128-16 layered structure followed by ReLU activation and BN in each layer. We used the Monte Carlo method to draw samples from $q_\phi(\mathbf{z})$ and used a standard Gaussian for $p(\mathbf{z})$.

For the DANN, we set $\alpha = 0.1$ in Eq. 6, whereas for the VDANN, we set $\alpha = 0.1$ and $\beta = 0.1$. For the InfoVDANN, we set $\alpha = 0.1$, $\beta = 1.0$, $\eta = 0.2$, and $\lambda = 1.0$.

5.2. PLDA training and scoring

We used the standard Gaussian PLDA backend for scoring. For SRE16, the baseline PLDA model was trained on the augmented SRE04–10 data. For SRE18, Mixer6 and its augmentation were also added to the training sets. The augmentation step follows Kaldi's SRE16 recipe. Before PLDA training, the x-vectors were projected to a 150 dimensional space by an LDA transformation matrix, followed by length normalization. The LDA projection matrix was trained on the same dataset as for training the PLDA models.

We also applied the PLDA adaptation detailed in the Kaldi's SRE16 recipe as an *extra* adaptation. Specifically, SRE16 unlabeled data were used to adapt the PLDA model for SRE16, while we used SRE18 unlabeled data for PLDA adaptation for SRE18.

For the evaluations of InfoVDANN, VDANN and DANN, we applied the same preprocessing as the baseline except that the transformed x-vectors were used for centering, LDA training, PLDA training, adaptation and scoring.

6. RESULTS AND DISCUSSIONS

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

Table 2 shows the pooled evaluation performance of the systems on SRE16. We can observe that both MMD-VDANN and AAE-VDANN consistently outperform the VDANN, while MMD-VDANN achieves the best performance. The right part presents the results using Kaldi's PLDA adaptation as an *extra* adaptation. We see that Kaldi's PLDA adaptation is still powerful because even though the x-vectors have been transformed by the InfoVDANNs, they outperform the baseline by a small margin only. The results in Table 2 show that both InfoVDANNs benefit DA in extracting additional speaker discriminative information from the training data compared with the VDANN, and that incorporating an MI compensation is effective for reducing domain mismatch.

Performance on SRE18-CMN2 is shown in Table 3. We obtain similar conclusions as in SRE16: Maintaining high MI between the latent features and the input can feed more speaker information into learned embeddings, which enhances speaker recognition performance.

The P-values of the McNemar's test [36] between VDANN and InfoVDANN are all zeros for SRE16 and SRE18. This means that the improvement of InfoVDANNs over VDANN is statistically significant.

Table 2. Performance on SRE16

	No PLDA adaptation		PLDA adaptation	
	EER	minDCF	EER	minDCF
Baseline	11.30	0.890	8.27	0.604
DANN	11.62	0.862	8.43	0.599
VDANN	11.13	0.845	8.22	0.585
MMD-VDANN	**10.74**	**0.825**	**7.87**	**0.575**
AAE-VDANN	10.90	0.834	7.96	0.579

Table 3. Performance on SRE18-CMN2

	No PLDA adaptation		PLDA adaptation	
	EER	minDCF	EER	minDCF
Baseline	11.21	0.676	9.60	0.575
DANN	10.79	0.678	9.31	0.584
VDANN	10.24	0.667	9.22	0.578
MMD-VDANN	**9.95**	**0.653**	**8.97**	**0.568**
AAE-VDANN	10.08	0.661	8.99	0.569

7. CONCLUSIONS

In this paper, we proposed an InfoVDANN for unsupervised DA. InfoVDANN incorporates an InfoVAE into the DANN to encourage higher MI between learned features and the input, while simultaneously retaining the advantage of VDANN as a Gaussian distribution regularizer. Experimental results on SRE16 and SRE18-CMN2 show that InfoVDANN is capable of reducing domain mismatch. The fact that the InfoVDANN consistently outperforms VDANN suggests that feeding suitable MI in training InfoVDANN is effective for extracting *extra* useful information for SV.

8. REFERENCES

[1] S.H. Shum, D.A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adap-

tation in speaker recognition systems," in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 266–272.

[2] L. Li and M.W. Mak, "Unsupervised domain adaptation for gender-aware PLDA mixture models," in *Proc. ICASSP*, 2018, pp. 5269–5273.

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

[4] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proc. ICASSP*, 2014, pp. 4002–4006.

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

[6] W. Lin, M.W. Mak, and J.T. 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.

[7] W. Lin, M.W. Mak, Y. Tu, and J.T. Chien, "Semi-supervised nuisance-attribute networks for domain adaptation," in *Proc. ICASSP*, 2019, pp. 6236–6240.

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

[9] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[10] 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.

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

[12] Q. Wang, W. Rao, S. Sun, L. Xie, E.S. Chng, and H. Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *Proc. ICASSP*, 2018, pp. 4889–4893.

[13] J. Rohdin, T. Stafylakis, A. Silnova, H. Zeinali, L. Burget, and O. Plchot, "Speaker verification using end-to-end adversarial language adaptation," in *Proc. ICASSP*, 2019, pp. 6006–6010.

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

[15] 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. Interspeech*, 2018, pp. 72–76.

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

[17] Y. Tu, M.W. Mak, and J.T. Chien, "Variational domain adversarial learning for speaker verification," in *Proc. Interspeech*, 2019, pp. 4315–4319.

[18] D.P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *ICLR*, 2014.

[19] Y. Zhang, L. Li, and D. Wang, "VAE-based regularization for deep speaker embedding," in *Proc. Interspeech*, 2019, pp. 4020–4024.

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

[21] A. Alemi, B. Poole, I. Fischer, J. Dillon, R.A. Saurous, and K. Murphy, "Fixing a broken ELBO," in *Proc. ICML*, 2018, pp. 159–168.

[22] J. He, D. Spokoyny, G. Neubig, and T. Berg-Kirkpatrick, "Lagging inference networks and posterior collapse in variational autoencoders," in *ICLR*, 2019.

[23] 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.

[24] D.J. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. ICML*, 2015, pp. 1530–1538.

[25] S.R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. CoNLL*, 2016, pp. 10–21.

[26] J.M. Tomczak and M. Welling, "VAE with a VampPrior," in *Proc. AISTATS*, 2018, pp. 1214–1223.

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

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

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

[30] C.P. 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.

[31] Y. Kim, S. Wiseman, A.C. Miller, D. Sontag, and A.M. Rush, "Semi-amortized variational autoencoders," in *Proc. ICML*, 2018, pp. 2678–2687.

[32] R. Shu, H.H. Bui, S. Zhao, M.J. Kochenderfer, and S. Ermon, "Amortized inference regularization," in *Proc. NIPS*, 2018, pp. 4393–4402.

[33] T.R. Davidson, L. Falorsi, N.D. Cao, T. Kipf, and J.M. Tomczak, "Hyperspherical variational auto-encoders," in *Proc. Uncertainty in Artificial Intelligence*, 2018, pp. 856–865.

[34] A.B. Dieng, Y. Kim, A.M. Rush, and D.M. Blei, "Avoiding latent variable collapse with generative skip models," in *Proc. AISTATS*, 2019, pp. 2397–2405.

[35] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

[36] L. Gillick and S.J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. ICASSP*, 1989, pp. 532–535.