Bayesian Learning for Domain-Invariant Speaker Verification and Anti-Spoofing

Jin Li[1,2], *Man-Wai Mak*[1], *Johan Rohdin*[2], *Kong Aik Lee*[1], *Hynek Hermansky*[2]

[1]Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong SAR
[2]Speech@FIT, Brno University of Technology, Czechia
jin666.li@connect.polyu.hk

Abstract

The performance of automatic speaker verification (ASV) and anti-spoofing drops seriously under real-world domain mismatch conditions. The relaxed instance frequency-wise normalization (RFN), which normalizes the frequency components based on the feature statistics along the time and channel axes, is a promising approach to reducing the domain dependence in the feature maps of a speaker embedding network. We advocate that the different frequencies should receive different weights and that the weights' uncertainty due to domain shift should be accounted for. To these ends, we propose leveraging variational inference to model the posterior distribution of the weights, which results in Bayesian weighted RFN (BWRFN). This approach overcomes the limitations of fixed-weight RFN, making it more effective under domain mismatch conditions. Extensive experiments on cross-dataset ASV, cross-TTS anti-spoofing, and spoofing-robust ASV show that BWRFN is significantly better than WRFN and RFN.

Index Terms: domain generalization, Bayesian learning, speaker verification, anti-spoofing

1. Introduction

Automatic speaker verification (ASV) seeks to authenticate a speaker's identity by analyzing their voice [1, 2]. It is widely applied across various applications, including biometric authentication on personal smart devices [3]. Recently, inspired by the robust feature extraction abilities of deep neural networks (DNNs), numerous deep learning-based speaker recognition methods have been introduced [4].

However, ASV systems deployed in real-world applications are negatively affected by domain mismatch between the training and test conditions due to various factors, such as variations in recording devices, acoustic environment, audio channels, and languages. In addition, ASV systems are vulnerable to unseen spoofing attacks, which are often created by new algorithms that are not seen by the systems during the training phase [5]. Domain adaptation (DA) [6] and domain generalization (DG) [7] are two mainstream approaches to address the domain mismatch problem.

DA has been developed to address domain shift problems by transferring knowledge from a well-labeled, resource-rich source domain to a target domain [8–10]. For example, a domain adversarial training approach addresses the domain mismatch problem by projecting various domains into the same subspace [9], thereby becoming domain invariant across seen domains. A probabilistic linear discriminant analysis (PLDA) adaptation was used to cluster unlabeled in-domain data and then use this data to adapt the parameters of out-of-domain models [10]. Correlation alignment (CORAL) was another widely used backend method that aligns out-of-domain statistics with those of the in-domain statistics [8].

On the other hand, domain generalization learns a model that remains robust to domain shift without requiring adaptation to the target domains [11]. Numerous approaches have been proposed to address the domain generalization issue by achieving strong generalization to unseen test domains in recent years. For example, the meta-generalized speaker verification was introduced via meta-learning to improve the generalization ability of the model on unseen domains [12]. In [13], a mutual information-based embedding decoupling method was proposed to improve domain generalization capabilities. More recently, Relaxing Instance Frequency-wise Normalization (RFN) was incorporated explicitly into the model to eliminate instance-specific domain discrepancies in acoustic features while minimizing the loss of useful discriminative information [14].

However, previous works are still prone to overfitting and poor generalization because the use of fixed parameters during inference fails to account for model uncertainty, leading to overfitting on the source domains [15]. Although it was shown in [14] that frequency-wise distribution is highly correlated with domain information, the approach suffers from overfitting on the source domain because the uncertainty of weights for layer normalization and instance frequency-wise normalization is not taken into account.

In this paper, we address domain generalization by extending prior work in [14] under a probabilistic framework. To better explore domain-invariant learning, we model the uncertainty in the weighting parameters of RFN by leveraging variational Bayesian inference. Bayesian learning has been applied to solve many machine learning problems in speech community [16–18]. It was applied to model the uncertainty on different normalizers to cope with the model's uncertainty. To better explore domain-invariant learning, we introduce uncertainty to the weights of the normalized frequency components in the convolutional feature maps of a speaker embedding network by leveraging variational Bayesian inference. This enables us to explore domain invariance in a principled way to achieve domain-invariant feature representations and speaker classification jointly. In our experiments, we directly compare the performance of a Bayesian learning-based neural network with a non-Bayesian learning-based neural network. Additionally, we evaluate our methods against various baselines in multiple tasks, including ASV, anti-spoofing, and SASV.

2. Preliminaries

In this section, we define the notations and present the preliminary on *relaxed instance frequency-wise normalization*.

2.1. Relaxed Instance Frequency-wise Normalization

An audio signal processing system typically applies spectral-domain analysis on each audio channel on a frame-by-frame basis. Previous studies on audio features [14] have shown that the frequency features (vectors whose elements are the frequency components) contain more domain-relevant information than the vectors defined by the channel axis. Therefore, domain mismatch could be reduced by reducing the domain-dependent variability along the frequency axis. One approach to reducing this variability is to normalize the frequency components in a channel-independent manner using the statistics obtained from the entire audio signal. The authors in [14] named the method "Instance Frequency-wise Normalization (INF)."

Given a mini-batch of N multichannel audio signals, its audio characteristics in the frequency domain can be represented by a tensor $\boldsymbol{x} \in \mathbb{R}^{N\times C\times F\times T}$, where N, C, F, and T are the mini-batch size, numbers of channels, frequency bins, and frames, respectively. The IFN of $\boldsymbol{x}$ at channel c and frame t is defined as:

$$\text{IFN}(\boldsymbol{x})_{:,c,:,t} = \frac{\boldsymbol{x}_{:,c,:,t} - \mathbb{E}_{\text{IFN}}(\boldsymbol{x})}{\sqrt{\mathbb{V}_{\text{IFN}}(\boldsymbol{x}) + \epsilon}} \in \mathbb{R}^{N\times F}\ , \tag{1}$$

where $\mathbb{E}_{\text{IFN}}$ and $\mathbb{V}_{\text{IFN}}$ are the first and second order statistics given by

$$\mathbb{E}_{\text{IFN}}(\boldsymbol{x}) = \frac{1}{C\cdot T}\sum\nolimits_c^C \sum\nolimits_t^T \boldsymbol{x}_{:,c,:,t} \tag{2}$$

and

$$\begin{aligned}\mathbb{V}_{\text{IFN}}(\boldsymbol{x}) = \frac{1}{C\cdot T}\sum\nolimits_c^C \sum\nolimits_t^T &(\boldsymbol{x}_{:,c,:,t} - \mathbb{E}_{\text{IFN}}(\boldsymbol{x})) \\ &\odot (\boldsymbol{x}_{:,c,:,t} - \mathbb{E}_{\text{IFN}}(\boldsymbol{x}))\ ,\end{aligned} \tag{3}$$

respectively. Here, ϵ is a small positive constant, and $\odot$ represents the Hadamard product. Define the layer normalization (LN) at channel c, frequency f, and frame t as:

$$\text{LN}(\boldsymbol{x})_{:,c,f,t} = \frac{\boldsymbol{x}_{:,c,f,t} - \mathbb{E}_{\text{LN}}(\boldsymbol{x})}{\sqrt{\mathbb{V}_{\text{LN}}(\boldsymbol{x}) + \epsilon}} \in \mathbb{R}^{N}\ , \tag{4}$$

where $\mathbb{E}_{\text{LN}}(\boldsymbol{x}) = \frac{1}{C\cdot F\cdot T}\sum_c^C \sum_f^F \sum_t^T \boldsymbol{x}_{:,c,f,t}$, and $\mathbb{V}_{\text{LN}}(\boldsymbol{x}) = \frac{1}{C\cdot F\cdot T}\sum_c^C \sum_f^F \sum_t^T (\boldsymbol{x}_{:,c,f,t} - \mathbb{E}_{\text{LN}}(\boldsymbol{x})) \odot (\boldsymbol{x}_{:,c,f,t} - \mathbb{E}_{\text{LN}}(\boldsymbol{x}))$. Then, the Relaxed instance Frequency-wise Normalization (RFN) is calculated as follows:

$$F(\boldsymbol{x}) = \lambda\text{LN}(\boldsymbol{x}) + (1-\lambda)\text{IFN}(\boldsymbol{x}) \in \mathbb{R}^{N\times C\times F\times T}\ , \tag{5}$$

where the scalar $\lambda \in [0,1]$ denotes the degree of relaxation.

3. Bayesian Relaxed Instance Frequency-wise Normalization

Eq. 5 interpolates between IFN and LN style normalization. However, the degree of relaxation, λ, of individual frequency bins are fixed and equal. We hypothesize that different frequency bins have different levels of domain dependence and should be weighted differently. We therefore extend RFN to weighted RFN (or WRFN), as follows

$$\begin{aligned}F(\boldsymbol{x}) =& \lambda\text{LN}(\boldsymbol{x}) \odot \sigma(\boldsymbol{w}_1) + (1-\lambda)\text{IFN}(\boldsymbol{x}) \odot \sigma(\boldsymbol{w}_2) \\ &\in \mathbb{R}^{N\times C\times F\times T}\ ,\end{aligned} \tag{6}$$

where $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^F$ are broadcast along the mini-batch, time and channel axes, $\sigma(\cdot)$ is a sigmoid function that squashes the weights to values between 0 and 1, WLN is weighted LN, and WIFN is weighted IFN. We define $\boldsymbol{w} = [\boldsymbol{w}_1^\mathsf{T}, \boldsymbol{w}_2^\mathsf{T}]^\mathsf{T}$, where T is the transpose. Note that the frequency specific weights, $\lambda\sigma(\boldsymbol{w}_1)$ and $\lambda\sigma(\boldsymbol{w}_2)$ generally does not sum to one. However, this can to a large extent be compensated by the subsequent convolutionary layer.

The weights in WRFN, $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$, are deterministic and learned from training data. Since the purpose of these weights is to address domain variations and since there number of domains in the training data is scarce, we hypothesize that the weights will be prone to overfitting. To mitigate this, we improve the robustness of the weighted RFN (Eq. 6) by modeling the uncertainty in $\boldsymbol{w}$ with Bayesian techniques, which results in Bayesian Weighted Relaxed Instance Frequency-wise Normalization (BWRFN).

To account for uncertainty in $\boldsymbol{w}$, we assume that the weight vector is drawn from a prior distribution $p(\boldsymbol{w})$. Let $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^N$ denote the *training set*, considering of N utterances, each spoken by one of the S speakers. As defined in Eqs. 1–6, each tensor $\boldsymbol{x}_i \in \mathbb{R}^{C\times F\times T}$ contains the acoustic features (e.g., filterbank features) of an utterance, and the corresponding speaker ID is represented as a one-hot vector $\boldsymbol{s}_i \in \mathbb{R}^S$. To predict the speaker label (or generate a speaker embedding) for a *test utterance*, we require the posterior distribution $p(\boldsymbol{w}|\mathcal{D})$ along with the other model parameters. Since computing the posterior $p(\boldsymbol{w}|\mathcal{D})$ is intractable, we employ variational inference (see e.g. Ch. 10 of [19]) to approximate it. Specifically, $p(\boldsymbol{w}|\mathcal{D})$ is approximated by a Gaussian distribution $q(\boldsymbol{w})$ with mean $\boldsymbol{\mu}$ and diagonal covariance $\boldsymbol{\sigma}^2$. The KL divergence, $D_{KL}(q(\boldsymbol{w})||p(\boldsymbol{w}|\mathcal{D}))$, can be minimized jointly with maximization of the (conditional) marginal log-likelihood of the training data, $p\left(\{\boldsymbol{s}_i\}_{i=1}^N|\{\boldsymbol{x}_i\}_{i=1}^N\right)$, by maximizing the variational lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma})$ as described in [20]:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = &\overbrace{\sum_{i=1}^N \mathbb{E}_{\boldsymbol{w}\sim q(\boldsymbol{w})} \log p_{\boldsymbol{\theta}}(\boldsymbol{s}_i|\boldsymbol{x}_i, \boldsymbol{w})}^{\mathcal{L}_1} \\ &- D_{KL}(q(\boldsymbol{w})||p(\boldsymbol{w})).\end{aligned} \tag{7}$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{s}_i|\boldsymbol{x}_i, \boldsymbol{w})$ is the speaker probabilities given by softmax layer of the neural network and $\boldsymbol{\theta}$ are the parameters of the neural network (other than $\boldsymbol{w}$). The prior, $p(\boldsymbol{w})$ is assumed to be a standard Gaussian, i.e., $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{0}, \boldsymbol{I})$. On the other hand, the variational posterior $q(\boldsymbol{w})$ is modeled as a diagonal Gaussian distribution, i.e., $q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$, where the covariance matrix is defined as $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \text{diag}(\boldsymbol{\sigma}_{\boldsymbol{w}}^2)$. The diagonal Gaussian distribution is initialized randomly but learned during training to approximate the true posterior $p(\boldsymbol{w}|\mathcal{D})$. The first term of Eq. 7 is the expectation of log-likelihood of the data over the approximated posterior distribution $q(\boldsymbol{w})$. This expectation can be computed by using Monte Carlo sampling, i.e.,

$$\begin{aligned}\mathcal{L}_1 &= \sum_{i=1}^N \int_{\boldsymbol{w}} \log p_{\boldsymbol{\theta}}(\boldsymbol{s}_i|\boldsymbol{x}_i, \boldsymbol{w}) q(\boldsymbol{w})\,\mathrm{d}\boldsymbol{w} \\ &\approx \frac{1}{K}\sum_{k=1}^K \sum_{i=1}^N \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{s}_i|\boldsymbol{x}_i, \boldsymbol{w}^{(k)})\right]\ ,\end{aligned}$$

where K denotes the number of samples and $\boldsymbol{w}^{(k)}$ is the k-th sample drawn from the distribution $q(\boldsymbol{w})$.

For sampling $\boldsymbol{w}^{(k)}$, the re-parametrization trick [20] is applied to produce the i-th element $w_i^{(k)}$ of $\boldsymbol{w}^{(k)}$ as follows:

$$w_i^{(k)} = \mu_{\boldsymbol{w},i} + \sigma_{\boldsymbol{w},i}\epsilon_i^{(k)}, \;\; \epsilon_i^{(k)} \sim \mathcal{N}(0,1)\,, \tag{8}$$

where $\mu_{\boldsymbol{w},i}$ is the i-th element of $\boldsymbol{\mu}_{\boldsymbol{w}}$ and $\sigma_{\boldsymbol{w},i}$ is the square root of the i-th diagonal element of $\boldsymbol{\Sigma}_{\boldsymbol{w}}$.

By assuming Gaussians for $p(\boldsymbol{w})$ and $q(\boldsymbol{w})$, the second term of Eq. 7 has a closed-form solution. It can be derived as follows:

$$\begin{aligned} \mathcal{L}_{\text{KL}} &= -D_{KL}(q(\boldsymbol{w})||p(\boldsymbol{w})) \\ &= -D_{KL}(q(\boldsymbol{w})|\mathcal{N}(\boldsymbol{w};\boldsymbol{0},\boldsymbol{I})) \\ &= \frac{1}{2}\sum_{f=1}^{F}\left[1+\log(\sigma_{\boldsymbol{w},f}^2) - \sigma_{\boldsymbol{w},f}^2 - \mu_{\boldsymbol{w},f}^2\right]\,, \end{aligned} \tag{9}$$

where $\sigma_{\boldsymbol{w},f}$ and $\sigma_{\boldsymbol{w},f}$ are the parameters of the variational posterior, and f represents index of frequency bins F.

During inference, posterior predictive uncertainty consists of aleatoric and epistemic components. Epistemic uncertainty stems from the limited amount of domains used to estimate the model, while aleatoric uncertainty arises from the inherent randomness in the data. The posterior predictive distribution:

$$\begin{aligned} \hat{\boldsymbol{y}} &= \int_{\boldsymbol{w}} p_{\boldsymbol{\theta}^*}(\hat{y}(\boldsymbol{z})|\boldsymbol{w})p(\boldsymbol{w}|\mathcal{D})\,\mathrm{d}\boldsymbol{w} \\ &= \mathbb{E}_{\boldsymbol{w}\sim p(\boldsymbol{w}|\mathcal{D})}\left[p_{\boldsymbol{\theta}^*}(\hat{y}(\boldsymbol{z})|\boldsymbol{w})\right]\,, \end{aligned} \tag{10}$$

where $\hat{y}(\cdot)$ represents the embedding layer output, $\boldsymbol{\theta}^*$ is an optimal parameters of the neural network (other than $\boldsymbol{w}$), and $\boldsymbol{z}$ is the Mel-spectrogram of a new utterance. This distribution can be expressed as the expectation of the single network likelihood under the posterior $p(\boldsymbol{w}|\mathcal{D})$, interpreting the predictive distribution as an infinite ensemble of network's outputs [21]. Each network's output contribution is weighted by the posterior of the weights given to the training data. This infinite ensemble can be approximated using a finite number of Monte Carlo samples from the posterior. For simplicity, we use the expected value of embedding layer output during inference.

4. Experimental Setup

4.1. Evaluation Tasks

The experiments were conducted on cross-dataset ASV tasks, anti-spoofing tasks, and spoofing-robust automatic speaker verification (SASV) tasks.

4.1.1. Cross-Dataset Speaker Verification

We followed [12] for cross-dataset speaker verification. To assess robustness, we designed a setup using CN-Celeb [22] (testing genre mismatches), HI-MIA [23] (reflecting device and environment variability), FFSVC [24] (containing cross-channel far-field audio), and Voxceleb [25] with 1,307 training speakers in total. Our trials simulate cross-genre and cross-device conditions to comprehensively evaluate genre, device, and dataset mismatches. During training, HI-MIA was treated as the unseen domain, while the model was trained on the remaining datasets.

4.1.2. Cross-TTS Anti-spoofing

In practice, attackers may employ various TTS, voice conversions (VC), and adversarial methods to generate spoofed speech, leading to domain mismatches between the training, development, and evaluation sets. In the closed condition of ASVspoof 5 Track 1 [26], participants were limited to using only the provided training and development data. Following [26], we built our systems on the training and development set and evaluated them on the evaluation set (see Table 2 of [26]).

4.1.3. Cross-TTS SASV

ASVspoof 5 Track 2 [26] evaluates spoofing-robust ASV (SASV) systems by simulating a telephony scenario where synthetic and converted speech was directly injected into a telephone line. Participants may develop either standalone classifiers or fuse separate ASV and countermeasure (CM) subsystems.

4.2. Implementation Details for Cross-dataset Speaker Verification

We used data augmentation to train speaker embedding networks, including adding music, noise, and babble from MUSAN [27] and convolving the original speech with the room impulse responses from RIR [28]. The original and augmented utterances were then cut into 2-second segments. For each segment, we extracted 40-dimensional mel-filterbank features using a 25ms window with a frameshift of 10ms. The filterbank features were then presented to different front-ends, as described below.

- *R-vector.* The R-vector network [29, 30] employs ResNet-18 to transform the mel-filterbank features into deep features, which were then mapped to 256-dimensional embeddings via average pooling. During training, softmax was used to compute the cross-entropy loss. The model was trained for 100 epochs using SGD with a momentum of 0.9, a weight decay of 0.0001, a mini-batch size of 100, and a learning rate decayed every 10 epochs from 0.1.
- *ECAPA-TDNN.* We also used ECAPA-TDNN [31] to extract speaker embeddings, using cross-entropy loss (softmax) to optimize the network.
- *RFN-R-vector.* We constructed an RFN-R-vector network by inserting an RFN operation (Eq. 5) before the first convolution layer and after each residual block of the R-vector network. All other training settings remain the same as those in the R-vector, with λ in Eq. 5 set to 0.5 as in [14].
- *WRFN-R-vector.* A WRFN-R-vector network is obtained by weighting the normalized frequency components (Eq. 6) in an RFN-R-vector network for each RFN operation.
- *BWRFN-R-vector.* We built a BWRFN-R-vector network by integrating Bayesian learning into the frequency normalization weights (Eqs. 8–9) of the WRFN-R-vector network while keeping the same training settings as the RFN-vector network. For efficiency, a single Monte Carlo sample was used for each 4-second speech segment (i.e., $K = 1$ in Eq. 8, similar to the setting in [32].). Additionally, we replaced the WRFN operations in the WRFN-R-vector network with BWRFN operations.

4.3. Implementation Details for Cross-TTS Anti-spoofing

BWRFN-ResNet was based on ResNet18 [33] with the addition of BWRFN, where the placement of BWRFN was consistent with that in *BWRFN-R-vector*. We applied extensive data augmentation during training, using MUSAN's [27] noise subset, RIR [28], RawBoost [34], Audiomentations,[1] and codecs [35].

4.4. Implementation Details for Cross-TTS SASV

Following [33], the ASVspoof 5 dataset was used for training. A ResNet34 generated ASV log-likelihood ratios (LLRs) for speaker verification, while the *BWRFN-ResNet* model produced countermeasure (CM) LLRs for anti-spoofing. The SASV system fuses both LLRs to compute the final SASV LLR, using the fusion and calibration process from [33].

5. Experiments and Analysis

This section presents the experimental results and analysis of domain generalization performance. For cross-dataset ASV evaluation, the EER is reported after the final training epoch. For the closed condition of Track 1, both minDCF and EER were used as evaluation metrics, while the closed condition of Track 2 was assessed using min a-DCF, min t-DCF, and t-EER.

Table 1: *EERs (%) of the proposed method and other baselines on the cross-dataset ASV task. WRFN-R-vector represents weighted RFN-R-vector, where the weights were optimized during training. A comprehensive evaluation is conducted by combining both seen and unseen trials into a unified set of assessment trials, referred to as "Overall".*

Methods	Seen		Unseen	Overall
	FFSVC	CN-Celeb	HI-MIA	
R-vector	4.56	16.01	12.65	13.36
ECAPA-TDNN	9.67	23.04	12.30	16.84
RFN-R-vector [14]	4.39	16.30	12.96	13.64
WRFN-R-vector	4.26	**16.08**	14.16	13.05
BWRFN-R-vector(ours)	**4.24**	16.96	**8.15**	**12.38**

Table 1 presents the performance evaluations across various datasets. Multiple datasets with diverse domain shifts are utilized to better simulate real-world complexities. The results show that the BWRFN-R-vector achieves the lowest EER in the seen FFSVC and unseen HI-MIA. This improvement stems from the BWRFN-R-vector's ability to model more intricate relationships between different normalizers. These findings highlight that the Bayesian-based weight modeling in BWRFN-R-vector enhances both the robustness and accuracy of speaker verification, particularly for out-of-distribution datasets like HI-MIA. Additionally, our method outperforms WRFN-R-vector, especially in unseen domains. This demonstrates that Bayesian learning methods surpass learnable weighted methods in cross-dataset evaluation.

Table 2: *EERs (%) of our proposed method and state-of-the-art methods on the cross-dataset ASV task. We inserted the BWRFN into different residual blocks (L1—L4) of the R-vector.*

Method	Overall
R-vector	13.36
R-vector+BWRFN-L1(ours)	12.83
R-vector+BWRFN-L2(ours)	**12.38**
R-vector+BWRFN-L3(ours)	12.58
R-vector+BWRFN-L4(ours)	12.96

We conduct experiments to find which layer is better for inserting the BWRFN operations, and the results are reported in Table 2. The varying performance across the residual blocks suggests that the position of BWRFN integration within the R-vector framework can influence the overall effectiveness, with L2 being the optimal residual block for this particular task.

Table 3: *Results of BWRFN-Resnet on the development set of the closed condition of Track 1 of ASVspoof 5.*

Methods	minDCF	EER
SincNet-ASSIST [36]	0.35	13.71
ResNet-18 [37]	0.20	12.15
ResNet-34 [37]	0.18	11.84
binspf [33]	0.14	12.16
BWRFN-ResNet (ours)	**0.13**	**11.36**

Table 3 shows that the performance of our proposed method achieves superior performance on anti-spoofing compared with other state-of-the-art methods. This improvement highlights the effectiveness of our method in addressing domain mismatch for different unknown TTS algorithms. The good performance is attributed to the ability of Bayesian learning, which can account for the parameter uncertainty and generalize better for the unseen domain.

Table 4: *Results of BWRFN-Resnet on the development set of the closed condition of Track 2 of ASVspoof 5.*

Methods	min a-DCF	min t-DCF	t-EER
ECAPA-TDNN+ASSIST [38]	0.1692	N/A	N/A
AASIST3 [39]	N/A	0.2657	N/A
FwSE-ResNet100 [40]	0.134	0.219	6.53
CM#5,ASV#2 [33]	0.127	0.209	6.57
BWRFN-ResNet (ours)	**0.125**	**0.205**	**5.83**

Our method outperforms the current state-of-the-art approaches for the closed condition of Track 2 of ASVspoof 5, as shown in Table 4. This demonstrates the effectiveness of our approach in tackling the cross-domain problem and highlights its advantages in domain generalization. Compared with other methods, our model achieves significant improvements, reflecting its ability to better capture domain-invariant features.

6. Conclusions and Future Work

In this paper, we have introduced the BWRFN-R-vector network, which integrates Bayesian learning into the weighting of the frequency components in the relaxed instance frequency-wise normalization. Results demonstrate that accounting for the uncertainty in the frequency-dependent weights can mitigate domain generalization issues. In future work, we plan to extend this approach to other front-end models.

7. Acknowledgements

This work was supported by the RGC of Hong Kong SAR, Grant No. PolyU 15228223 and the PolyU's Research Student Attachment Programme. The work was supported by Czech Ministry of Education, Youth and Sports (MoE) through the OP JAK project "Linguistics, Artificial Intelligence and Language and Speech Technologies: from Research to Applications" (ID:CZ.02.01.01/00/23_020/0008518). Computing on IT4I supercomputer was supported by MoE through the e-INFRA CZ (ID:90254).

[1] https://github.com/iver56/audiomentations

8. References

[1] M.-W. Mak and J.-T. Chien, *Machine learning for speaker recognition*. Cambridge University Press, 2020.

[2] J. Li, M.-W. Mak, N. Yan, and L. Wang, "Modeling suprasegmental information using finite difference network for end-to-end speaker verification," in *Proc. APSIPA ASC*, 2023, pp. 119–124.

[3] K. A. Lee, B. Ma, and H. Li, "Speaker verification makes its debut in smartphone," *SLTC Newsletter*, February 2013.

[4] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[5] C. Zeng, X. Miao, X. Wang, E. Cooper, and J. Yamagishi, "Spoofing-aware speaker verification robust against domain and channel mismatches," in *Proc. STL*, 2024, pp. 1150–1157.

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[7] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 4396–4415, 2022.

[8] K. A. Lee, Q. Wang, and T. Koshinaka, "The CORAL+ algorithm for unsupervised domain adaptation of PLDA," in *Proc. ICASSP*, 2019, pp. 5821–5825.

[9] 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.

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

[11] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proce. the AAAI*, 2018, pp. 3490–3497.

[12] H. Zhang, L. Wang, K. A. Lee, M. Liu, J. Dang, and H. Meng, "Meta-generalization for domain-invariant speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1024–1036, 2023.

[13] J. Li, J. Han, S. Deng, T. Zheng, Y. He, and G. Zheng, "Mutual information-based embedding decoupling for generalizable speaker verification," in *Proc. Interspeech*, 2023, pp. 3147–3151.

[14] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Proc. Interspeech*, 2022, pp. 2393–2397.

[15] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[16] M. W. Lam, S. Hu, X. Xie, S. Liu, J. Yu, R. Su, X. Liu, and H. Meng, "Gaussian process neural networks for speech recognition." in *Proc. Interspeech*, 2018, pp. 1778–1782.

[17] B. Xue, S. Hu, J. Xu, M. Geng, X. Liu, and H. Meng, "Bayesian neural network language modeling for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2900–2917, 2022.

[18] X. Li, J. Zhong, J. Yu, S. Hu, X. Wu, X. Liu, and H. Meng, "Bayesian x-vector: Bayesian neural network based x-vector system for speaker verification," in *Proc. Odyssey*, 2020, pp. 365–371.

[19] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. Springer, 2006, vol. 4.

[20] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," *Proc. NIPS*, pp. 2575–2583, 2015.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. ICML*, 2015, pp. 1613–1622.

[22] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "CN-Celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[23] X. Qin, H. Bu, and M. Li, "Hi-mia: A far-field text-dependent speaker verification database and the baselines," in *Proc. ICASSP*, 2020, pp. 7609–7613.

[24] X. Qin, M. Li, H. Bu, W. Rao, R. K. Das, S. Narayanan, and H. Li, "The interspeech 2020 far-field speaker verification challenge," *arXiv preprint arXiv:2005.08046*, 2020.

[25] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[26] X. Wang, H. Delgado, H. Tak, J. weon Jung, H. jin Shim, M. Todisco, I. Kukanov, X. Liu, M. Sahidullah, T. H. Kinnunen, N. Evans, K. A. Lee, and J. Yamagishi, "ASVspoof 5: crowdsourced speech data, deepfakes, and adversarial attacks at scale," in *Proc. ASVspoof 2024*, 2024, pp. 1–8.

[27] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[28] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, 1979.

[29] X. Qin, D. Cai, and M. Li, "Far-field end-to-end text-dependent speaker verification based on mixed training data with transfer learning and enrollment data augmentation." in *Proc. Interspeech*, 2019, pp. 4045–4049.

[30] S. Wang, Y. Yang, Z. Wu, Y. Qian, and K. Yu, "Data augmentation using deep generative models for embedding based speaker recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2598–2609, 2020.

[31] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech*, 2020, pp. 3830–3834.

[32] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[33] J. Rohdin, L. Zhang, P. Oldřich, V. Staněk, D. Mihola, J. Peng, T. Stafylakis, D. Beveraki, A. Silnova, J. Brukner, and L. Burget, "BUT systems and analyses for the ASVspoof 5 Challenge," in *Proc. ASVspoof*, 2024, pp. 24–31.

[34] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *Proc. ICASSP*, 2022, pp. 6382–6386.

[35] A. Okhotnikov, I. Yakovlev, N. Torgashov, R. Makarov, E. Gómez, P. Malov, A. Alenin, and A. Balykin, "IDVoice team system description for ASVSpoof5 Challenge," in *Proc. ASVspoof*, 2024, pp. 43–47.

[36] P. Falez and T. Marteau, "Whispeak speech deepfake detection systems for the ASVspoof5 Challenge," in *Proc. ASVspoof*, 2024, pp. 32–35.

[37] A.-T. Dao, M. Rouvier, and D. Matrouf, "ASVspoof 5 Challenge: advanced ResNet architectures for robust voice spoofing detection," in *Proc. ASVspoof*, 2024.

[38] O. Kurnaz, S. C. Demirtaş, A. B. J. Mishra, and C. Hanilçi, "Spoofing-robust speaker verification using parallel embedding fusion: BTU speech group's approach for ASVspoof5 Challenge," in *Proc. ASVspoof*, 2024, pp. 138–143.

[39] K. Borodin, V. Kudryavtsev, D. Korzh, A. Efimenko, G. Mkrtchian, M. Gorodnichev, and O. Y. Rogov, "AASIST3: KAN-enhanced AASIST speech deepfake detection using SSL features and additional regularization for the ASVspoof 2024 Challenge," in *Proc. ASVspoof*, 2024, pp. 48–55.

[40] J. A. Villalba, T. Feng, T. Thebaud, J. Lee, S. Narayanan, and N. Dehak, "The SHADOW team submission to the ASVSpoof 2024 Challenge," in *Proc. ASVspoof*, 2024, pp. 36–42.