Interspeech 2024
1-5 September 2024, Kos, Greece

Challenging margin-based speaker embedding extractors by using the variational information bottleneck

Themos Stafylakis[1,2], *Anna Silnova*[3], *Johan Rohdin*[3], *Oldrich Plchot*[3], *Lukas Burget*[3]

[1]Athens University of Economics and Business, Greece
[2]Omilia - Conversational Intelligence, Athens, Greece
[3]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia

tstafylakis@aueb.gr, isilnova@fit.vutbr.cz

Abstract

Speaker embedding extractors are typically trained using a classification loss over the training speakers. During the last few years, the standard softmax/cross-entropy loss has been replaced by the margin-based losses, yielding significant improvements in speaker recognition accuracy. Motivated by the fact that the margin merely reduces the logit of the target speaker during training, we consider a probabilistic framework that has a similar effect. The variational information bottleneck provides a principled mechanism for making deterministic nodes stochastic, resulting in an implicit reduction of the posterior of the target speaker. We experiment with a wide range of speaker recognition benchmarks and scoring methods and report competitive results to those obtained with the state-of-the-art Additive Angular Margin loss.

Index Terms: speaker recognition, variational information bottleneck

1. Introduction

During the last several years, speaker embeddings extracted with speaker-discriminatively trained deep neural networks have attained impressive performance on several datasets. During this period, the field has witnessed numerous architectures, such as LSTM-based models ([1]), 1D (TDNN, ECAPA-TDNN, xi-vectors, TitaNet [2, 3, 4, 5]) and 2D CNNs (ResNet, ResNeXt, Res2Net [6, 7, 8]), and more recently self-supervised pretrained Transformer models (Wav2Vec 2.0, HuBERT, WavLM [9, 10]), fined-tuned for the particular downstream task.

One of the most effective and architecture-agnostic methods to improve speaker embedding discriminability has been the margin-based loss, especially the Additive-Angular Margin (AAM [11, 12]) variant. Originating from face recognition, AAM combines directional statistics with a margin penalty on the angle between the embedding and the prototype of the target class. It comes as a drop-in replacement of regular softmax/cross-entropy loss and achieves substantial improvements in many speaker recognition benchmarks.

On the other hand, the effectiveness of margin in other, more challenging datasets is questionable. Margin-based losses do not retain much of their strength when a backend model (e.g., probabilistic linear discriminant analysis, PLDA) needs to be introduced in the pipeline [13, 14]. This is crucial since most of the successful systems in NIST-SRE do make use of such a backend. Furthermore, in industry-level voice biometrics, where a single extractor typically serves several different deployments (often in different languages), a backend model trained on in-domain data is usually compulsory for attaining state-of-the-art performance. Finally, margin-based losses are not based on a probabilistic framework, which is a desired property for tasks and methods like calibration, self-supervised training, and multitask learning.

In this paper, we propose an alternative to margin-based loss, based on the variational information bottleneck (VIB [15]). The method has been introduced in deep learning for some years, and although it has been applied to many fields (e.g., speech antispoofing, computer vision, NLP, explainability, a.o. [16, 17, 18, 19, 20]), it has not been sufficiently explored for training speaker embedding extractors. In [21], an approach similar to ours is presented, however, the experiments are conducted only with VoxCeleb1, which is considered low-resource, with limited test-set, and having low speaker variability compared to, e.g., NIST-SRE or CNCeleb. Furthermore, to the best of our knowledge, we are the first to underscore the similarities between VIB and margin-based losses, conduct experiments on challenging benchmarks such as NIST-SRE, and combine it with backend models (e.g., PLDA).

In the rest of the paper, we examine whether making the embedding stochastic and sampling from its conditional distribution during training may have a similar impact on the speaker-discriminability of the embeddings. We experiment with a wide collection of speaker recognition benchmarks, namely VoxCeleb [22, 23], CNCeleb [24], and the latest NIST-SREs [25, 26, 27]. Moreover, we examine different backend methods, ranging from pure cosine-scoring to PLDA and Toroidal Probabilistic Spherical Discriminant Analysis (T-PSDA) [28]. Finally, we provide arguments in favor of further exploration of VIB and propose future research directions.

2. Training the network with VIB

In this section, the method for training the embedding extractor with VIB is demonstrated. We begin by providing some intuition of the Information Bottleneck principle and the derivation of VIB. We then discuss similarities of VIB with other methods, while providing implementation details of our method.

2.1. The Information Bottleneck principle

The Information Bottleneck [29] builds on the desired properties of a model utilizing some internal representation: such representation has to be effective at performing the task the model is used for; at the same time, the information about the input data contained in such representation should be as compressed as possible. In IB, these general considerations are formalized by introducing the following maximization problem:

$$R_{IB} = I(Z, Y; \boldsymbol{\theta}) - \beta I(Z, X; \boldsymbol{\theta}), \tag{1}$$

10.21437/Interspeech.2024-2058

where the second term corresponds to the mutual information between the input X and its internal representation Z that should be minimized, and the first one is the mutual information between Z and the output Y to be maximized. The scalar $\beta > 0$ controls the amount of compression of X in Z, while $\boldsymbol{\theta}$ is a vector of model parameters.

2.2. Derivation of VIB

In practice, for an arbitrary model, estimating mutual information terms in eq. (1) can be challenging. Deep Variational Information Bottleneck (VIB) [15] addresses this issue by introducing two variational approximations to the IB by parameterizing it as a neural network. The first is $q(y|z;\boldsymbol{\psi})$, which is an approximation to the true conditional distribution $p(y|z)$, where the former is parametrized by the decoder's parameters $\boldsymbol{\psi}$, which in our case corresponds to the linear layer that maps Z to the logits (i.e., it is a collection of speaker prototypes, one for each training speaker). The second variational approximation is $r(z)$, with which we approximate the marginal distribution of Z, i.e., $r(z) \approx p(z) = \int p(z|x;\boldsymbol{\theta})p(x)dx$. Although the parameters of $r(z)$ can be learnable, it is common to keep them fixed and equal to those of a standardized multivariate normal (MVN), i.e., $r(z) = \mathcal{N}(z|0, I)$ – the same is used in the experiments of this paper.

Using the above variational approximations, we obtain

$$I(Z, Y;\boldsymbol{\theta}) \geq \int p(x)p(y|x)p(z|x;\boldsymbol{\theta}) \log q(y|z;\boldsymbol{\psi})dxdydz \tag{2}$$

which is equal to the negative expected cross-entropy, where the expectation is with respect to the training data $p(x)p(y|x)$ but also with respect to $p(z|x;\boldsymbol{\theta})$.

Similarly, $I(Z, X;\boldsymbol{\theta})$ is upper-bounded by the following expression

$$I(Z, X;\boldsymbol{\theta}) \leq \int p(x)p(z|x;\boldsymbol{\theta}) \log \frac{p(z|x;\boldsymbol{\theta})}{r(z)}dxdz \tag{3}$$

which is equal to the expected Kullback-Leibler (KL) divergence between $p(z|x;\boldsymbol{\theta})$ and $r(z)$, with respect to the training data distribution $p(x)$. Therefore, R_{IB} is lower-bounded by the following expression

$$\begin{aligned}
R_{IB} &\geq \mathbb{E}_{p(x)p(y|x)p(z|x;\boldsymbol{\theta})}[\log q(y|z;\boldsymbol{\psi})] \\
&\quad - \beta\mathbb{E}_{p(x)p(z|x;\boldsymbol{\theta})}\left[\log \frac{p(z|x;\boldsymbol{\theta})}{r(z)}\right] \\
&= -\mathbb{E}_{p(x)p(z|x;\boldsymbol{\theta})}[\text{CE}(p(y|x), q(y|z;\boldsymbol{\psi})] \\
&\quad - \beta\mathbb{E}_{p(x)}[\text{KL}(p(z|x;\boldsymbol{\theta}), r(z))] \\
&\approx \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m}\sum_{j=1}^{m}\left(\log q(y_i|z^{(j)};\boldsymbol{\psi})\right)\right. \\
&\quad \left. - \beta\text{KL}(p(z|x_i;\boldsymbol{\theta}), r(z))\right]
\end{aligned} \tag{4}$$

2.3. Sampling from the conditional distribution via the reparametrization trick

Similarly to variational autoencoders (VAEs [30]), VIB models $p(z|x;\boldsymbol{\theta})$ by a MVN with diagonal covariance matrix $\mathcal{N}(z|f_\mu(x), \text{diag}(f_\sigma^2(x)))$. Therefore, the KL term in eq. (4) for a given x is simply the KL divergence between two MVN distributions. For calculating the CE term in eq. (4) for each given x, samples from $p(z|x;\boldsymbol{\theta})$ are required. The reparametrization trick of VAEs is utilized, where samples $e^{(j)}$ are generated from a standardized MVN and transformed to samples $z^{(j)}$ from the target distribution $p(z|x;\theta)$ as follows

$$z^{(j)} = f_\sigma(x) \odot e^{(j)} + f_\mu(x),\ e^{(j)} \sim \mathcal{N}(0, I). \tag{5}$$

where $j = 1, \dots, J$. We implement $f_\mu(x)$ and $f_\sigma(x)$ with two linear layers having the statistics pooling layer as common input. Note that statistics pooling is defined as mean and std pooling for each frequency-channel pair of the feature maps of the last convolutional layer. These mean and std pooled features should not be confused with $f_\mu(x)$ and $f_\sigma(x)$, as the latter correspond to the mean and std parametrizing the distribution of the stochastic embedding Z. The non-negativity of the diagonal elements of $f_\sigma(x)$ is enforced by the softplus function, although the exponential function may also be applied. Finally, the encoder is a ResNet-34 with statistics pooling layer, although other architectures may be considered.

2.4. VIB and VAE

As we observe, VIB and VAE have several similarities, since (a) they both make use of a stochastic internal representation Z, (b) they prevent the collapse of $p(z|x;\boldsymbol{\theta})$ to a point mass by penalizing the KL divergence between it and $p(z)$, and (c) they make use of the reparametrization trick to sample from $p(z|x)$. From this perspective, one may consider VIB as a supervised learning analog of VAEs. On the other hand, there are certain differences, the main of which are the following.

- The classifier-decoder of the VIB, as opposed to the (deep) reconstruction-based decoder of VAE. Note that this allows generating many samples $z^{(j)}$ (i.e., $J \gg 1$) per training example to reduce the variance of the estimator with a minor increase in computation since the classifier is merely a linear layer (we set $J = 10$).
- The information-theoretic formulation of VIB, as opposed to the variational-Bayes (VB) formulation of VAEs, which among others provides a better justification of the tunable trade-off parameter $\beta > 0$ (The VB formulation implies $\beta = 1$).

2.5. Similarities with margin-based losses

Apart from VAEs, VIB has certain similarities with margin-based losses. The AAM loss is a variant of Softmax/CE, where the logit of the ith training example and jth training speaker is modified as $l_{i,j} = s\cos(\theta_{z_i,y_j} + m\delta_{j,y_i})$, where s is the scale (typically $s = 30$), m is the margin, $\delta_{\cdot,\cdot}$ is the Dirac function, and θ_{z_i,y_j} is the (positive) angle between the embedding z_i and the jth speaker prototype. It therefore makes the classification task artificially harder, by intervening to the logits and reducing the one corresponding to the target class. This intervention creates a safety zone around the class boundaries, which increases the discriminability of the speaker embeddings, especially for speakers not appearing in the training set.

In the VIB framework, Z is assumed stochastic, its conditional distribution $p(z|x;\boldsymbol{\theta})$ is estimated and samples are generated from it, which should be classified using regular Softmax/CE. Having to classify samples from $p(z|x;\boldsymbol{\theta})$ instead of the expected value $\mathbb{E}_{p(z|x;\boldsymbol{\theta})}[Z]$ makes the task harder, as it increases the within-class variability of the embeddings. This results from the fact that $\mathbb{E}_{p(z|x;\boldsymbol{\theta})} \log q(y|z;\boldsymbol{\psi}) \leq \log q(y|\mathbb{E}_{p(z|y;\boldsymbol{\psi})}[Z];\boldsymbol{\psi})$ where y is the ground-truth speaker. As sampling is applied only during training, it creates a safety zone around the class boundaries, similarly to the AAM loss.

Note that other regularization techniques are employed for making the task harder during training, such as data augmentation (by adding noise and reverberation or applying SpecAugment), short training utterances, or dropout. Nonetheless, the similarity between VIB and margin stems from the fact that they both act on the classifier of the architecture, while the other methods are either input-level or act on the intermediate layers. Therefore, as with margin-based losses, the other regularization techniques are orthogonal to VIB.

2.6. Angular margin, unit-length normalization, and VIB

Margin-based losses are typically combined with angular distances, e.g., by unit-length normalizing the embedding and the class/speaker prototypes. For example, penalizing the target logit with a fixed additive margin without applying such unit-length normalizations may not be effective, as the network would be free to overcome the penalty by increasing the average magnitude of the embeddings, and therefore increasing the average scale of the logits, rendering the margin ineffective. As shown in Sect. 2.5, the AAM unit-length normalizes both the embeddings and speaker prototypes and bounds the target logit to $[-s, s\cos(m)]$.

On the other hand, VIB does not necessitate unit-length normalization of neither the embedding nor the prototypes. As discussed in Sect 2.2, the KL term enforces the aggregated conditional distribution $p(z) = \int p(z|x;\boldsymbol{\theta})p(x)dx$ of the training set to be as close as possible to $r(z)$ preventing the magnitude (length) of Z from increasing beyond the (soft) boundaries of the support of $r(z)$. This inherent mechanism of VIB provides us with flexibility in choosing whether or not to length-normalize embeddings and/or prototypes, e.g., depending on the benchmark and the scoring method to be used.

3. Experiments

We perform three sets of experiments, NIST-SRE, VoxCeleb, and CNCeleb, to analyze the performance of VIB regularization. All of them were conducted using WeSpeaker ([31]) toolkit[1] and closely followed experimental setups of the corresponding recipes including training set, augmentations, and training hyperparameters like optimizer, learning rate, etc. The only exception is that we used longer training examples of 300 frames (instead of the default 200) in all experiments. Also, we use our custom implementation of scoring backends for NIST-SRE experiments. When implementing the training with VIB we reuse the same scheduler as used for the margin in WeSpeaker: for the first 20 epochs, the margin (or β in VIB) is set to zero and then exponentially increased to its final value in the course of the next 20 epochs after which it is kept constant for the rest of the training.

3.1. NIST-SRE

In these experiments, we train all embedding extractors on NIST CTS superset [32] and test their performance on evaluation sets of three editions of NIST SRE: SRE2016 [25], telephone condition from SRE2018 [26], and audio-only part of SRE 2021 [27]. In all cases, the embeddings were centered and length-normalized. Then, we applied linear discriminant analysis (LDA) reducing the dimensionality of the embeddings from 256 to 100. Finally, we trained two scoring backends on preprocessed embeddings: PLDA and T-PSDA. When PLDA is used for scoring, both speaker and channel subspaces have a dimensionality of 100 (i.e., we use the two-covariance version of the PLDA model); when T-PSDA is used, we do not attempt to tune its hyperparameters but rather adopt the same values as found optimal in [28]: we use one 60-dimensional speaker variable and two 5-dimensional channel variables. The parameters for centering, LDA, and the backends were estimated on the embedding extractor training set. The performance is reported in terms of equal error rate (EER) and minimum cost ($\min\_C$) as defined by the respective evaluation plan and computed by the scoring tools provided by the evaluation organizers.

Table 1 shows the comparison of the baseline embedding extractors trained with regular Softmax/CE or AAM objectives versus the VIB version of the same objective. In the first column of the table, the number in the brackets corresponds to the value of the margin m for AAM and to β for VIB. The table is separated into two parts for two different scoring backends allowing not only comparison between the backends themselves but also showing the effectiveness of the VIB approach across different scoring methods. By analyzing the results, we see that VIB consistently outperforms the corresponding loss not utilizing the margin (VIB vs. CE and VIB_LN vs. AAM(0.0)) while in most of the cases being competitive to the margin-based AAM(0.2).

3.2. VoxCeleb

When running the experiments on VoxCeleb dataset, we follow a commonly adopted setup: we use the development part of VoxCeleb2 [23] to train the embedding extractor and the whole VoxCeleb1 [22] for testing. As a backend, we use simple cosine scoring with only centering and length-normalization of the embeddings as preprocessing. The centering is done with the mean computed on VoxCeleb2 development set. The same set was used as a cohort for adaptive score normalization, where we selected 300 highest scores from the cohort to estimate normalization parameters.

Table 2 displays the results achieved with the baseline extractors trained with AAM with margins 0.2 and 0.0 along with the one that uses VIB. Similar to NIST-SRE case, we observe that VIB consistently outperforms AAM loss with the margin set to zero, although when the margin is used AAM is clearly superior. These observations are valid for both cases: whether we use score-normalization or not. Apart from this, the table shows the large-margin fine-tuned model along with an analogous setting for VIB: in both cases, the length of the training examples was increased to 6 seconds for the last 10 epochs with the margin increased to 0.5 and the value of β kept the same as for the rest of the training (0.004 in this case).

3.3. CNCeleb

Following WeSpeaker CNCeleb recipe, we use a combination of CNCeleb2 and the development set of CNCeleb1 as the training set and evaluate on the test set of CNCeleb1. For multi-enrollment trials, the embeddings for all sessions are extracted and averaged to obtain a single enrollment embedding.

We performed a similar set of experiments to the ones presented for VoxCeleb dataset: we compared the performance of several embedding extractors trained with AAM or VIB with and without finetuning on the long training examples. In all cases, simple cosine scoring was used, with optionally using score normalization with the network training set used as a cohort. The results are given in Table 3 and show that VIB is competitive to AAM loss in most of the metrics.

[1] https://github.com/wenet-e2e/wespeaker

Table 1: *Results on NIST SRE evaluation sets.*

		SRE16 yue		**SRE16 tgl**		**SRE18 CMN2**		**SRE21 audio**	
loss	**backend**	**min_C**	**EER(%)**	**min_C**	**EER(%)**	**min_C**	**EER(%)**	**min_C**	**EER(%)**
CE	PLDA	.470	4.50	.998	20.17	.569	7.81	.564	10.54
VIB(0.002)	PLDA	.326	3.36	.951	14.79	.517	6.56	.555	10.50
AAM(0.0)	PLDA	.336	3.39	.988	15.19	.502	6.55	.552	10.83
AAM(0.2)	PLDA	.336	3.34	**.849**	**13.14**	**.487**	**6.19**	.562	10.29
VIB_LN(0.002)	PLDA	**.315**	**3.16**	.984	14.91	.491	**6.19**	**.538**	**9.42**
CE	T-PSDA	.441	4.80	.872	16.16	.569	7.96	.573	9.98
VIB(0.002)	T-PSDA	.394	4.00	.807	12.65	.541	6.55	**.565**	10.44
AAM(0.0)	T-PSDA	**.355**	3.86	**.742**	11.37	**.518**	6.48	.569	10.26
AAM(0.2)	T-PSDA	.399	3.96	.758	**10.87**	**.518**	6.30	.599	10.37
VIB_LN(0.002)	T-PSDA	.360	**3.64**	.779	12.12	.525	**6.29**	.588	**9.51**

Table 2: *Results on VoxCeleb with cosine scoring, without and with score normalization (wo/w).*

	Vox1-O		**Vox1-E**		**Vox1-H**	
	minDCF$_{0.01}$	**EER(%)**	**minDCF$_{0.01}$**	**EER(%)**	**minDCF$_{0.01}$**	**EER(%)**
AAM(0.0)	.150/.131	1.28/1.11	.154/.132	1.33/1.18	.245/.192	2.55/2.13
AAM(0.2)	.115/.090	0.96/0.83	.114/.104	0.98/0.89	.182/.160	1.86/1.63
VIB_LN(0.004)	.109/.094	0.99/0.88	.130/.115	1.11/1.02	.204/.174	2.08/1.82
AAM(0.2)+FT(0.5)	.074/.056	0.88/0.74	.101/.092	0.95/0.89	.173/.151	1.69/1.53
VIB_LN+FT(0.004)	.113/.078	0.96/0.80	.121/.108	1.06/0.95	.194/.160	1.96/1.71

Table 3: *Results on the CNCeleb evaluation set, with cosine scoring, without and with score normalization (wo/w).*

	minDCF$_{0.01}$	**EER(%)**
AAM(0.0)	.406/.367	7.97/7.25
AAM(0.2)	.394/.360	7.22/6.61
VIB_LN(0.004)	.406/.372	7.24/6.87
AAM(0.2)+FT(0.5)	.393/.356	7.23/6.56
VIB_LN+FT(0.004)	.399/.364	7.29/6.78

3.4. Discussion

The experiments we conducted on VoxCeleb and CNCeleb show that our implementation of VIB is comparable to the state-of-the-art AAM loss, recovering most of its performance gains compared to AAM with zero margin (i.e., only length-normalization of embeddings and prototypes). The experiments on NIST-SRE show that our implementation of VIB is competitive to the best setting for each evaluation test we examine.

We emphasize that there are numerous directions one may consider to boost the performance of the method, such as deeper and less mutually-coupled branches for estimating $f_\mu(x)$ and $f_\sigma(x)$, different $r(z)$ such as Gaussian Mixture Models or von Mises-Fisher distribution, and several optimization techniques that can be found in the rich literature of VIB and VAE. We should also mention that most of the hyperparameters we used for training are optimized for the AAM loss, leaving room for further improvement simply by hyperparameter optimization.

Furthermore, the solid probabilistic/information-theoretic framework of VIB facilitates research in many directions, such as (a) unsupervised domain adaptation and multi-domain training (by adapting or conditioning the marginal $p(z)$ to each domain), (b) uncertainty propagation (by propagating the uncertainly of the conditional distribution $p(z|x;\boldsymbol{\theta})$ in the probabilistic backend [33, 34]), (c) disentangled speaker embedding extractors (e.g., [35]) and finally, (d) incorporating the embedding extractor to more general architectures (e.g., diarization, target-speaker extractor, voice conversion, joint speaker and speech recognition architecture, a.o.). Joint architectures typically behave better with well-defined distributions and losses, and VIB is a step towards this direction.

4. Conclusions

In this paper, we examined the strength of VIB in training speaker embedding extractors. Motivated by the wide adoption of margin-based losses, their strength in boosting the performance of virtually any architecture, but also their weaknesses, we tried to address the question of whether or not the VIB can eventually replace margin-based losses. We experimented with a diverse collection of speaker recognition benchmarks and used different scoring methods. Our experiments show that our implementation of VIB yields improvement over margin-based loss in several NIST-SRE test sets, while it can recover much of the performance gains of the margin-based loss in VoxCeleb and CNCeleb. Finally, several directions to improve its performance are provided, together with other speaker-related applications and settings in which VIB may be beneficial.

5. Acknowledgements

The work was partially supported by Czech Ministry of Interior project No. VB02000060, 2024-2026 ”NABOSO” and Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90254).

6. References

[1] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, “End-to-end text-dependent speaker verification,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification,” in *Interspeech*, 2020.

[4] K. A. Lee, Q. Wang, and T. Koshinaka, “Xi-vector embedding for speaker recognition,” *IEEE Signal Processing Letters*, vol. 28, pp. 1385–1389, 2021.

[5] N. R. Koluguri, T. Park, and B. Ginsburg, “Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8102–8106.

[6] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, “But system description to voxceleb speaker recognition challenge 2019,” *arXiv preprint arXiv:1910.12592*, 2019.

[7] M. Rybicka, J. Villalba, P. Zelasko, N. Dehak, and K. Kowalczyk, “Spine2net: Spinenet with res2net and time-squeeze-and-excitation blocks for speaker recognition.” in *Interspeech*, 2021, pp. 496–500.

[8] T. Zhou, Y. Zhao, and J. Wu, “Resnext and res2net structures for speaker verification,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 301–307.

[9] S. Novoselov, G. Lavrentyeva, A. Avdeeva, V. Volokhov, and A. Gusev, “Robust speaker recognition with transformers using wav2vec 2.0,” *arXiv preprint arXiv:2203.15095*, 2022.

[10] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černocký, “Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[12] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, “Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition,” in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1652–1656.

[13] A. Silnova, T. Stafylakis, L. Mošner, O. Plchot, J. Rohdin, P. Matějka, L. Burget, O. Glembek, and N. Brümmer, “Analyzing speaker verification embedding extractors and back-ends under language and channel mismatch,” in *Odyssey 2022: The speaker and Language Recongnition Workshop, Beijing*, 2022. [Online]. Available: https://arxiv.org/abs/2203.10300

[14] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak *et al.*, “State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations,” *Computer Speech & Language*, vol. 60, p. 101026, 2020.

[15] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” in *International Conference on Learning Representations*, 2016.

[16] Y. Eom, Y. Lee, J. Um, and H.-R. Kim, “Anti-spoofing using transfer learning with variational information bottleneck,” in *23rd Annual Conference of the International Speech Communication Association, INTERSPEECH 2022*. ISCA, 2022, pp. 3568–3572.

[17] Y. Du, J. Xu, H. Xiong, Q. Qiu, X. Zhen, C. G. Snoek, and L. Shao, “Learning to learn with variational information bottleneck for domain generalization,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 200–216.

[18] S. Cui, J. Cao, X. Cong, J. Sheng, Q. Li, T. Liu, and J. Shi, “Enhancing multimodal entity and relation extraction with variational information bottleneck,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[19] T. Gu, G. Xu, and J. Luo, “Sentiment analysis via deep multichannel neural networks with variational information bottleneck,” *IEEE Access*, vol. 8, pp. 121 014–121 021, 2020.

[20] S. Bang, P. Xie, H. Lee, W. Wu, and E. Xing, “Explaining a blackbox by using a deep variational information bottleneck approach,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 13, 2021, pp. 11 396–11 404.

[21] D. Wang, Y. Dong, Y. Li, Y. Zi, Z. Zhang, X. Li, and S. Xiong, “Variational information bottleneck based regularization for speaker recognition.” in *Interspeech*, 2021, pp. 1054–1058.

[22] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech and Language*, 2019.

[23] J. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” *Interspeech 2018*, 2018.

[24] 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.

[25] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, “The 2016 nist speaker recognition evaluation,” *Interspeech 2017*, 2017.

[26] O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, “The 2018 nist speaker recognition evaluation.” INTERSPEECH, Graz, AT, 2019-09-15 00:09:00 2019. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=927673

[27] S. O. Sadjadi, C. Greenberg, E. Singer, L. Mason, and D. Reynolds, “The 2021 nist speaker recognition evaluation,” *arXiv preprint arXiv:2204.10242*, 2022.

[28] A. Silnova, N. Brümmer, A. Swart, and L. Burget, “Toroidal probabilistic spherical discriminant analysis,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[29] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *The 37th annual Allerton Conference on Communication, Control, and Computing*, 1999, pp. 368–377.

[30] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *In Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[31] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, “Wespeaker: A research and production oriented speaker embedding learning toolkit,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[32] O. Sadjadi, “NIST SRE CTS Superset: A large-scale dataset for telephony speaker recognition,” 2021-08-16 04:08:00 2021. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=933116

[33] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, “Plda for speaker verification with utterances of arbitrary duration,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7649–7653.

[34] Q. Wang, K. A. Lee, and T. Liu, “Incorporating uncertainty from speaker embedding estimation to speaker verification,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[35] T. Liu, K. A. Lee, Q. Wang, and H. Li, “Disentangling voice and content with self-supervision for speaker recognition,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.