Disentangled Representation Learning for Environment-agnostic Speaker Recognition

KiHyun Nam[1]*, Hee-Soo Heo*[2]*, Jee-weon Jung*[3]*, Joon Son Chung*[1]

[1]Korea Advanced Institute of Science and Technology, South Korea
[2]Naver Cloud Corporation, South Korea
[3]Carnegie Mellon University, USA

nkh.mmai@kaist.ac.kr, heesoo.heo@navercorp.com, jeeweonj@ieee.org, joonson@kaist.ac.kr

Abstract

This work presents a framework based on feature disentanglement to learn speaker embeddings that are robust to environmental variations. Our framework utilises an auto-encoder as a disentangler, dividing the input speaker embedding into components related to the speaker and other residual information. We employ a group of objective functions to ensure that the auto-encoder's code representation – used as the refined embedding – condenses only the speaker characteristics. We show the versatility of our framework through its compatibility with any existing speaker embedding extractor, requiring no structural modifications or adaptations for integration. We validate the effectiveness of our framework by incorporating it into two popularly used embedding extractors and conducting experiments across various benchmarks. The results show a performance improvement of up to 16%. We release our code for this work to be available here[1].

Index Terms: speaker recognition, disentangled representation learning, real environment, environment mismatch

1. Introduction

The growth of voice-based AI services has amplified the demand for robust speaker recognition models capable of operating effectively in noisy environments. Every audio recording carries not only the speaker-specific characteristics [1, 2], such as age, gender, accent [3], emotion [4,5] and language [6,7], but also environmental information [8] like noise and reverberation. These factors are intertwined within the speaker representation. While some of these factors are essential for identifying the speaker, others, particularly environmental information, can act as obtrusive information, making speaker recognition more challenging. This issue becomes more pronounced in an environment mismatch scenario, where changes in recording conditions – ranging from serene offices to bustling streets – can drastically alter audio characteristics, to the extent that they may seem to originate from different individuals. Consequently, the removal of these intrusive factors from speaker embeddings emerges as a pivotal step towards enhancing the effectiveness of speaker recognition systems, ensuring that they can distinguish between essential speaker-related information and environmental distortions.

Despite the use of datasets [9,10] recorded in various real-world environments and data augmentation techniques [11, 12], the emergence of realistic benchmark datasets [13–15] continually reveals the vulnerability of speaker recognition systems in varied environment conditions. This highlights the need for a more fundamental solution that can explicitly exclude environmental information from speaker representations.

Disentangled representation learning (DRL) [16] emerges as a reasonable approach for tackling this challenge. DRL seeks to independently isolate and manipulate the different factors within the input data, showing promise across various fields [2,7,17–21]. For example, [7] removes linguistic information from speaker representation using an adversarial-based DRL framework for bilingual speaking scenarios. DRL represents a promising approach, yet its occasional removal of vital information, which can compromise performance, highlights the need for continued refinement and exploration.

We propose a novel adversarial learning-based disentangled representation learning framework that can remove environmental information from speaker representations while minimising the loss of speaker information. Traditional adversarial learning-based DRL ensures effective information removal, but adversarial learning often distorts task-relevant information, resulting in training instability [7,22–24]. We introduce a new idea that uses an auto-encoder as a disentangler to separate environmental information while leveraging a reconstruction loss function of the auto-encoder to penalise unnecessary information loss, thereby mitigating the loss of vital speaker information during the DRL process. Additionally, we combine a set of objective functions to facilitate the learning of refined speaker information within the disentangled speaker embedding. To assist environment-DRL, we employ a regularisation technique that swaps embeddings of the same speaker from different environmental origins during the reconstruction process. Another contribution of our framework is the adaptability to seamlessly integrate with various existing speaker recognition networks without any structural modifications. Experimental results show that our framework demonstrates up to 16% performance improvement over baseline models and previous DRL framework on various evaluation sets.

We summarise our contributions as follows: (1) We introduce a novel DRL framework, which leverages an auto-encoder as a disentangler, to minimise the loss of vital information. (2) Our framework is easily adaptable to existing speaker networks without any structural modifications. (3) Our framework shows significant performance improvements on various evaluation sets reflecting in the wild conditions and also increases the performance of existing baseline models on standard benchmarks.

2. Related works

Triplet batch formulation. Our batch formulation is similar to those found in previous studies [25,26], which employed a triplet batch formulation – that is, each mini-batch index comprises three utterances. [25] constructed triplets of utterances from the same speaker by selecting two utterances from the same video and the third from a different, non-overlapping video. However, this approach did not incorporate further data augmentation, ensuring that the first two utterances were subjected to similar environmental noises, whereas the third was exposed to distinct noises. [26] also adopted a triplet batch formulation strategy but without utilising video session information, extracting all three non-overlapping

[1] Official webpage: https://mm.kaist.ac.kr/projects/voxceleb-disentangler/
Official code: https://github.com/kaistmm/voxceleb-disentangler

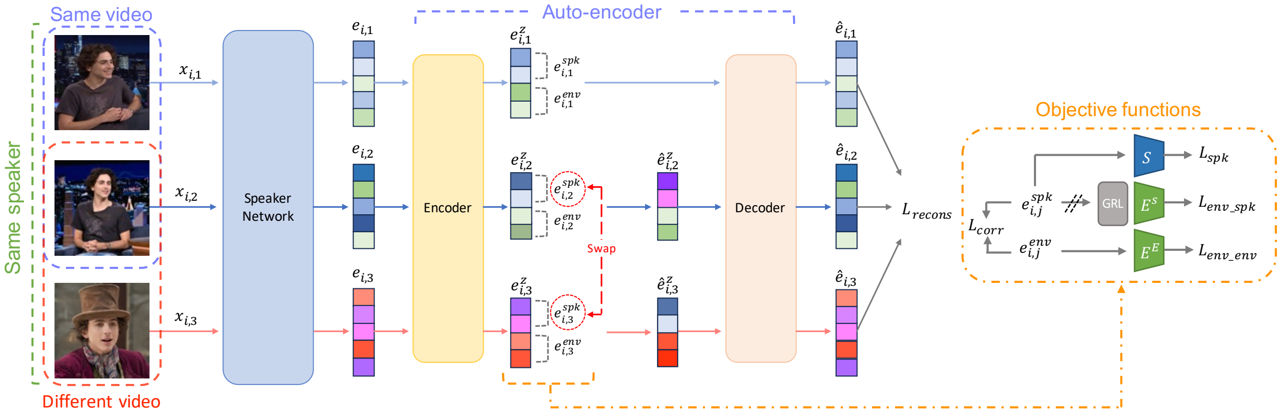

Figure 1: *The illustration of the proposed environment-disentangled representation learning framework. Auto-encoder encodes the speaker network's entangled speaker representation into a compact latent vector, which is then divided into distinct speaker and environment representation vectors. Orange box represents a set of objective functions to facilitate the learning of refined speaker and environment representations from the auto-encoder's bottleneck representation. Reconstruction training of the auto-encoder minimises the loss of vital speaker information during the disentangled representation learning.*

utterances from a single video and then simulating an environment mismatch scenario through artificial data augmentation.

Our strategy takes advantage of the studies above. We assemble the triplets in the same manner as [25] and additionally implement data augmentation techniques akin to those used by [26].

Feature enhancement using auto-encoder. Auto-encoders have been widely used to enhance latent embeddings for various purposes [18, 20, 27]. [27] introduced a method for enhancing speaker embeddings using an auto-encoder to reduce noise in speaker diarisation. However, this method did not specifically address the disentanglement of environmental noise, and the auto-encoder training was performed online within a diarisation framework using a single input recording. [18, 20] introduce a disentangling auto-encoder to reduce background information within audio signal and sign language embeddings. These studies proposed an embedding swapping technique to enhance the disentanglement capability.

Our work introduces an auto-encoder-based DRL framework that, for the first time, targets the disentanglement of environmental noise in speaker verification. Our proposed framework utilises an auto-encoder combined with a variety of objective functions. The framework incorporates adversarial learning and embedding swapping techniques designed to accurately disentangle environmental noise while preserving the speaker's fundamental characteristics.

3. Proposed disentanglement framework

This section presents the proposed environment DRL framework, as illustrated in Figure 1. Employing a set of triplets obtained through a specialised batch sampling method (Section 3.1), the auto-encoder condenses and reconstructs the input embeddings, as detailed in Section 3.2 (L_{recons}). Simultaneously, four additional objective functions (L_{env_env}, L_{env_spk}, L_{corr}, and L_{spk}) facilitate the training process to effectively isolate environmental noise while retaining essential speaker features, as discussed in Sections 3.3 and 3.4.

3.1. Batch construction with data augmentation

Our batch formulation is essentially identical to that of [25]. Each mini-batch index consists of three utterances: $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$, where i denotes the mini-batch index. All three utterances originate from the same speaker; however, the first two are sourced from an identical video, and the third is drawn from a different video. This setup aims to ensure that the first two utterances reflect the same environmental conditions, whereas the third introduces a distinct environment.

The novelty in batch construction stems from data augmentation. In contrast to [25], we apply identical augmentation techniques to the first two utterances and a different augmentation method to the third. This further ensures that the first two utterances share similar environmental noise while the third utterance involves distinct environmental noise.

3.2. Framework structure and reconstruction

Our disentanglement framework is independent of the speaker embedding extractor and corresponds to the block denoted as "Auto-encoder" in Figure 1. The framework inputs an arbitrary extractor's embedding, $e_{i,j} \in \mathbb{R}^D$where $e_{i,j}$ denotes the extracted speaker embedding from an input utterance $x_{i,j}, j \in \{1,2,3\}$. The encoder of the auto-encoder projects the input speaker representation $e_{i,j}$ into a compact representation $e^z_{i,j}$. The decoder reconstructs $\hat{e}_{i,j}$ using $e^z_{i,j}$ and a reconstruction loss calculates the L1 distance between the output $\hat{e}_{i,j}$ and the input $e_{i,j}$. The reconstruction loss L_{recons} is formulated as follows:

$$L_{recons} = \sum_{j=1}^{3}(|e_{i,j} - \hat{e}_{i,j}|). \qquad (1)$$

On top of the basic reconstruction loss of an auto-encoder introduced above, our framework employs several additional techniques and objective functions including adversarial learning loss to successfully train the model without collapsing or deteriorating.

3.3. Code swapping between different environments

Within the auto-encoder, the code $e^z_{i,j}$ is further split into a speaker representation $e^{spk}_{i,j} \in \mathbb{R}^d$ and an environment representation $e^{env}_{i,j} \in \mathbb{R}^{D-d}$ as shown in the middle part of Figure 1. Among a pair of triplets, we swap $e^{spk}_{i,2}$ with $e^{spk}_{i,3}$. This process guides the model to condense core speaker characteristics in $e^{spk}_{i,j}$ while environmental noise is projected to $e^{env}_{i,j}$. Note that we do not apply code swapping to $e^z_{i,1}$

3.4. Discriminator training

To learn fine-grained speaker and environment information within the two outputs from the encoder, $e_{i,j}^{spk}$ and $e_{i,j}^{env}$, respectively, we train a total of three discriminators: A speaker discriminator S and two environment discriminators E^E and E^S.

The speaker discriminator S computes the speaker classification loss L_{spk} from $e_{i,j}^{spk}$ to perform speaker recognition training. For L_{spk}, we employ a combination loss function, which is proposed in [28], with an angular prototypical loss [29] and a vanilla softmax loss. For the M value of the angular prototypical loss [29], we use $M=3$; one sample $e_{i,1}^{spk}$ from i-th triplet as a query set and other two samples $e_{i,2}^{spk}$ and $e_{i,3}^{spk}$ as a support set. For the vanilla softmax loss, the speaker discriminator S uses one fully-connected layer f to map $e_{i,j}^{spk}$ to a speaker class vector.

The environment discriminator E^E computes the environment loss function L_{env_env} by passing the environment-specific representation to an environment classifier g for environment recognition training. For the environment loss function, we employ a triplet loss as follows:

$$\begin{aligned} pos\_dist &= \left\| g(e_{i,1}^{env}) - g(e_{i,2}^{env}) \right\|_2^2 \\ neg\_dsit &= \left\| g(e_{i,1}^{env}) - g(e_{i,3}^{env}) \right\|_2^2 \\ L_{env\_env} &= max(0, m + pos\_dist - neg\_dist) \end{aligned} \tag{2}$$

where m is the margin of the triplet loss. The environment loss function ensures that the environment discriminator develops the ability to distinguish environment representations, making similar environment representations from the same video closer and those from different videos further apart.

3.5. Adversarial learning

The environment discriminator E^S performs to capture any residual environmental information from the disentangled speaker representation $e_{i,j}^{spk}$. E^S uses the same network structure and loss function as E^E, but they not share any parameters. $L_{\text{env\_spk}}$ is equal to Eq. 2, except that $L_{\text{env\_spk}}$ replaces the input $e_{i,j}^{env}$ with the $e_{i,j}^{spk}$. Since E^S should be trained independently, E^S is not connected to other neural networks, and the gradient is not propagated below the input data $e_{i,j}^{spk}$.

To explicitly remove residual environmental information from the speaker representation $e_{i,j}^{spk}$, we employ a combination of the gradient reversal layer (GRL) with the correlation minimisation loss proposed by [7]. The GRL is attached in front of the environment discriminator E^S, which inverts the gradient of the loss function L_{env_spk}, interfering with loss minimisation. By the GRL, our framework learns the speaker representation $e_{i,j}^{spk}$ that disrupts the environment discriminator E^S, thereby inducing the removal of the residual environmental information. We denote the loss function L_{env_spk} passing through the GRL as $L_{\text{env\_spk(G)}}$. As an additional regularisation, we employ the mean absolute pearson correlation (MAPC) loss, used in [7, 23], to minimise the correlation between speaker and environment representations. This loss function is denoted as L_{corr} and is formulated as follows:

$$L_{corr} = \frac{|\text{Cov}(e_{i,j}^{spk}, e_{i,j}^{env})|}{\sigma(e_{i,j}^{spk}) \cdot \sigma(e_{i,j}^{env})} \tag{3}$$

where $\text{Cov}(\cdot)$ and $\sigma(\cdot)$ mean the covariance and the standard deviation, respectively.
In summary, the overall loss function is defined as follows:

$$\begin{aligned} L_{total} = {} & \lambda_S * L_{spk} + \lambda_R * L_{recons} + \lambda_E * L_{env\_env} \\ & + \lambda_{adv} * L_{env\_spk(G)} + \lambda_C * L_{corr} \end{aligned} \tag{4}$$

where the lambda values are the weight values of losses summation. Same as the training process of [7], for the same mini-batch, L_{total} updates the framework modules excluding the environment discriminator E^S and L_{env_spk} updates only the E^S.

4. Experiments

4.1. Input representations

First, we randomly extract a 2-second audio segment from each utterance and apply pre-emphasis with a coefficient of 0.97. We transform the input signal into a spectrogram with a 25ms window size, 10ms stride size, and a hamming window and use it as input data for the speaker network. For the ResNet-34, we use a 64-dimensional log mel-spectrogram as the input data and applied instance normalisation [31] to the input. For the ECAPA-TDNN, we use 80-dimensional log mel-spectrogram as the input.

4.2. Model architecture

To demonstrate the compatibility of the proposed method on existing speaker recognition systems, we employ two existing models, the variant of ResNet-34 proposed in [28] and ECAPA-TDNN [30]. Note in this paper, we do not use the residual fully-connected layers following the pooling layer of both speaker networks. To compare our framework with the prior adversarial learning-based DRL model, we employ the GRL-based framework proposed in [7].

ResNet-34. We choose '**H / ASP**' version model mentioned in [28], which uses the attentive statistic pooling (ASP) [32] layer.

ECAPA-TDNN. ECAPA-TDNN [30] is a neural network, which consists of a series of 1-dimensional Res2Blocks. ECAPA-TDNN uses the channel- and context-dependent statistics pooling layer. We employ the large-size model used in [30].

Auto-encoder. The auto-encoder's encoder and decoder each consist of one batch normalisation layer followed by one fully-connected layer, sequentially. The input and output dimension sizes of the encoder and decoder are symmetrical. For the encoder, the input dimension size matches the output size of the speaker network's pooling layer, while the dimension size of the output vector e^z is 1024 and 512 for ResNet-34 and ECAPA-TDNN, respectively. The latent representation e^z is divided in half, equally split into the e^{spk} and the e^{env}. Before being passed to the decoder, we use L1 normalisation to both e^{spk} and e^{env} independently.

Discriminator. For the speaker discriminator S, we use just one fully-connected layer as f for the cross-entropy loss. Therefore, the output dimension size of the f is the same as the number of speaker classes. For the g of the environment discriminators E^E and E^S, we use two MLP layers and each MLP layer consists of a batch normalisation layer, an ELU [33] activation function, and a fully-connected layer, sequentially. For ResNet-34 and ECAPA-TDNN, the output sizes of the first MLP layer are 512 and 256, respectively, and the output sizes of the last MLP layer are 512 and 128, respectively.

4.3. Implementation details

Datasets. For training, we use the development sets from VoxCeleb2 [10]. Since the VoxCeleb datasets provide video session information for each speaker, we can utilise the video session information for the batch configuration described in Section 2.1. For evaluation, we select 6 multiple evaluation sets: three evaluation protocols utilising VoxCeleb1 [9], VoxSRC22 [15] and 23, and VC-Mix [34]. VoxSRC 22&23 and VC-Mix are the evaluation sets that reflect the environment mismatch problem. For data augmentation, we employ reverberations of simulated RIRs

Table 1: *EER and minDCF on (a) VoxSRC22&23 and VC-Mix evaluation sets, (b) VoxCeleb1-based evaluation sets. All experiments are repeated three times, and we report the mean and the standard deviation.* ***GRL*** *[7]: a prior work of the adversarial learning-based DRL framework using the gradient reversal layer;*

Model	**VoxSRC22**		**VoxSRC23**		**VC-Mix**	
	EER (%)	**minDCF**	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
ResNet-34 [28]	3.25 ± 0.041	0.211 ± 0.0013	5.91 ± 0.096	0.323 ± 0.0028	3.05 ± 0.091	0.245 ± 0.0051
+ *GRL* [7]	3.15 ± 0.101	0.209 ± 0.0086	5.60 ± 0.130	0.314 ± 0.0062	2.95 ± 0.207	0.253 ± 0.0157
+ ***Ours***	**2.95 ± 0.047**	**0.193 ± 0.0067**	**5.35 ± 0.126**	**0.306 ± 0.0024**	**2.58 ± 0.113**	**0.223 ± 0.0132**
ECAPA-TDNN [30]	3.25 ± 0.038	0.210 ± 0.0015	5.92 ± 0.016	0.328 ± 0.0018	2.92 ± 0.145	0.254 ± 0.0057
+ *GRL* [7]	3.22 ± 0.130	0.203 ± 0.0074	5.85 ± 0.105	**0.297 ± 0.0056**	2.62 ± 0.131	**0.211 ± 0.0089**
+ ***Ours***	**3.11 ± 0.065**	**0.199 ± 0.0075**	**5.81 ± 0.090**	0.325 ± 0.0006	**2.43 ± 0.059**	0.212 ± 0.0008

(a) *Results on VoxSRC22&23 evaluation sets and VC-Mix.*

Model	**Vox1-O**		**Vox1-E**		**Vox1-H**	
	EER (%)	**minDCF**	**EER (%)**	**minDCF**	**EER (%)**	**minDCF**
ResNet-34 [28]	0.95 ± 0.051	0.076 ± 0.0048	1.26 ± 0.028	0.089 ± 0.0022	2.51 ± 0.038	0.162 ± 0.0007
+ *GRL* [7]	1.13 ± 0.053	0.083 ± 0.0078	1.16 ± 0.035	0.081 ± 0.0019	2.34 ± 0.033	0.153 ± 0.0021
+ ***Ours***	**0.86 ± 0.024**	**0.068 ± 0.0047**	**1.10 ± 0.010**	**0.078 ± 0.0015**	**2.20 ± 0.016**	**0.142 ± 0.0031**
ECAPA-TDNN [30]	0.89 ± 0.024	0.072 ± 0.0093	1.16 ± 0.003	0.081 ± 0.0006	2.39 ± 0.003	**0.153 ± 0.0006**
+ *GRL* [7]	0.90 ± 0.046	0.076 ± 0.0030	1.19 ± 0.025	0.083 ± 0.0022	2.51 ± 0.035	0.163 ± 0.0024
+ ***Ours***	**0.82 ± 0.006**	**0.067 ± 0.0016**	**1.16 ± 0.011**	**0.080 ± 0.0021**	**2.38 ± 0.007**	0.156 ± 0.0006

(b) *Results on VoxCeleb1-based evaluation sets.*

dataset [12] and noises from the MUSAN dataset [11].

Training. All experiments are based on the PyTorch framework [35] and open-source `voxceleb_trainer`[2]. We use mixed precision training and the Adam Optimizer [36] with an initial learning rate of 0.001. ResNet-34-based experiments have a batch size of 220 and reduce the learning rate by 25% every 16 epochs. ECAPA-TDNN-based experiments have a batch size of 256 and decrease the learning rate by 25% every 8 epochs. Our implementation is performed on a single NVIDIA RTX 4090 GPU with 24 GB memory. Only the value of λ_{adv} is 0.5 and the values of other λ are 1. The training takes around 300 epochs. For all statistic pooling layers in the baseline and our models, all mean pooling parts are replaced with l_2-norm pooling [37].

Evaluation. We use three datasets, VoxSRC22&23 and VC-Mix, as evaluation sets to measure the environment robustness performance. Since the three datasets, Vox1-O, Vox1-E, and Vox1-H, are not designed to be dependent on a specific factor, we evaluate the generalisation performance of these three datasets. We measure the performances by two metrics: 1) the Equal Error Rate (EER), where the rates of False Rejections (FRR) and False Alarms (FAR) are identical, and 2) the minimum Detection Cost Function (minDCF) described in NIST SRE [38], which is a weighted sum of FRR and FAR. For minDCF, we use the parameters $C_{miss} = 1$, $C_{fa} = 1$ and $P_{target} = 0.05$. We sample each utterance into ten segments of 4 seconds each and compute the similarity across all segment pair combinations. The average similarity score is used as the trial's final score. This scoring is mentioned in [28].

5. Results

Our experimental results are summarised in Table 1. We compare three versions of each model, baseline, using only GRL [7] and using our framework. For reliable measurements, we train all models three times with different random seeds and report the mean and the standard deviations. We use the standard deviation values to measure the training stability of models. Table 1a demonstrates the performances under wild environmental condition evaluation sets. Table 1b shows the performances for VoxCeleb1-based evaluation sets to investigate the adaptability of our framework.

Environment-disentangled representation. As observed in Table 1a, both baseline models exhibit the lowest performances on wild environment evaluation sets, revealing vulnerabilities to the environment mismatch problem. In contrast, the models applying our framework achieve the best performances on the same evaluation sets, showing up to approximately 16% performance improvement over the baselines. This proves that our framework's ability to extract speaker information more clearly and strengthen independence from environmental factors. The models utilising only the GRL [7] also achieved performance improvements, but the more remarkable results across all evaluation sets by our framework confirm a higher capacity to exclude environmental information.

Generalisation. As shown in Table 1, the models employing only the GRL [7] show the highest standard deviation values across almost all evaluation sets except for one experiment with ResNet-34 on Vox1-H. Additionally, Table 1b reveals a performance degradation on the VoxCeleb1-based evaluation sets. This highlights our claim that such direct adversarial learning-based DRL without any constraint can lead to training instability and a failure in generalisation. Conversely, our framework not only reduces the standard deviation even though employing GRL but also shows approximately 12% improvement in the performance of baseline models, as illustrated in Table 1b. This proves that our framework's auto-encoder effectively mitigates information loss during the DRL process, facilitating generalisation through DRL.

6. Conclusion

We introduce a novel adversarial learning-based DRL framework for environment robust speaker recognition. Our framework leverages an auto-encoder as a disentangler to separate a speaker representation and an environment representation from an originally entangled speaker representation of a speaker network. The auto-encoder also works to minimise unnecessary loss of vital speaker representation through reconstruction training, and consequently, the proposed framework reduces the training instability caused by adversarial learning. The proposed framework shows significant performance improvement on evaluation sets that reflect varied environments and also on standard benchmarks.

[2] `https://github.com/clovaai/voxceleb_trainer`

7. Acknowledgements

This work was supported by the National Research Foundation of Korea grant funded by the Korean government (Ministry of Science and ICT, RS-2023-00212845) and the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00259991) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

8. References

[1] C. Luu, P. Bell, and S. Renals, “Leveraging Speaker Attribute Information Using Multi Task Learning for Speaker Verification and Diarization,” in *Proc. Interspeech*, 2021, pp. 491–495.

[2] C. Luu, S. Renals, and P. Bell, “Investigating the contribution of speaker attributes to speaker separability using disentangled speaker representations,” in *Proc. Interspeech*, 2022.

[3] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, “Probing the information encoded in x-vectors,” in *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 726–733.

[4] J. Williams and S. King, “Disentangling style factors from speaker representations.” in *Proc. Interspeech*, 2019, pp. 3945–3949.

[5] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, “x-vectors meet emotions: A study on dependencies between emotion and speaker recognition,” in *Proc. ICASSP*. IEEE, 2020, pp. 7169–7173.

[6] S. Maiti, E. Marchi, and A. Conkie, “Generating multilingual voices using speaker space translation based on bilingual speaker data,” in *Proc. ICASSP*. IEEE, 2020, pp. 7624–7628.

[7] K. Nam, Y. Kim, J. Huh, H. S. Heo, J.-w. Jung, and J. S. Chung, “Disentangled representation learning for multilingual speaker recognition,” in *Proc. Interspeech*, 2022.

[8] J. P. Campbell, “Speaker recognition: A tutorial,” *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[9] 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.

[10] J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep Speaker Recognition,” in *Proc. Interspeech*, 2018, pp. 1086–1090.

[11] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.

[12] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *Proc. ICASSP*. IEEE, 2017, pp. 5220–5224.

[13] A. Nagrani, J. S. Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, “VoxSRC 2020: The second VoxCeleb speaker recognition challenge,” *arXiv preprint arXiv:2012.06867*, 2020.

[14] A. Brown, J. Huh, A. Nagrani, J. S. Chung, and A. Zisserman, “Playing a part: Speaker verification at the movies,” in *Proc. ICASSP*. IEEE, 2021, pp. 6174–6178.

[15] J. Huh, A. Brown, J.-w. Jung, J. S. Chung, A. Nagrani, D. Garcia-Romero, and A. Zisserman, “VoxSRC 2022: The fourth voxceleb speaker recognition challenge,” *arXiv preprint arXiv:2302.10248*, 2023.

[16] X. Wang, H. Chen, S. Tang, Z. Wu, and W. Zhu, “Disentangled representation learning,” *arXiv preprint arXiv:2211.11695*, 2022.

[17] M. Sang, W. Xia, and J. H. Hansen, “Deaan: Disentangled embedding and adversarial adaptation network for robust speaker representation learning,” in *Proc. ICASSP*. IEEE, 2021, pp. 6169–6173.

[18] Y. Jang, Y. Oh, J. Cho, D. Kim, J. S. Chung, and I. S. Kweon, “Signing outside the studio: Benchmarking background robustness for continuous sign language recognition,” in *Proc. BMVC*. BMVA Press, 2022, p. 322.

[19] D. Yang, S. Huang, H. Kuang, Y. Du, and L. Zhang, “Disentangled representation learning for multimodal emotion recognition,” in *Proc. ACM MM*, 2022, pp. 1642–1651.

[20] A. Omran, N. Zeghidour, Z. Borsos, F. de Chaumont Quitry, M. Slaney, and M. Tagliasacchi, “Disentangling speech from surroundings with neural embeddings,” in *Proc. ICASSP*, 2023, pp. 1–5.

[21] S. Hao, K. Han, and K.-Y. K. Wong, “Learning attention as disentangler for compositional zero-shot learning,” in *Proc. CVPR*, 2023, pp. 15 315–15 324.

[22] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *Proc. ICLR*, 2017.

[23] W. H. Kang, S. H. Mun, M. H. Han, and N. S. Kim, “Disentangled speaker and nuisance attribute embedding for robust speaker verification,” *IEEE Access*, vol. 8, pp. 141 838–141 849, 2020.

[24] Y.-C. Wang, C.-Y. Wang, and S.-H. Lai, “Disentangled representation with dual-stage feature learning for face anti-spoofing,” in *Proc. WACV*, 2022, pp. 1955–1964.

[25] J. S. Chung, J. Huh, and S. Mun, “Delving into VoxCeleb: environment invariant speaker recognition,” in *Proc. Speaker Odyssey*, 2020.

[26] J. Kang, J. Huh, H. S. Heo, and J. S. Chung, “Augmentation adversarial training for self-supervised speaker representation learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1253–1262, 2022.

[27] Y. J. Kim, H.-S. Heo, J.-w. Jung, Y. Kwon, B.-J. Lee, and J. S. Chung, “Advancing the dimensionality reduction of speaker embeddings for speaker diarisation: disentangling noise and informing speech activity,” in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.

[28] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, “The ins and outs of speaker recognition: lessons from VoxSRC 2020,” in *Proc. ICASSP*. IEEE, 2021, pp. 5809–5813.

[29] J. S. Chung, J. Huh, S. Mun, M. Lee, H.-S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, “In Defence of Metric Learning for Speaker Recognition,” in *Proc. Interspeech*, 2020, pp. 2977–2981.

[30] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” in *Proc. Interspeech*, 2020.

[31] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.

[32] K. Okabe, T. Koshinaka, and K. Shinoda, “Attentive Statistics Pooling for Deep Speaker Embedding,” in *Proc. Interspeech*, 2018, pp. 2252–2256.

[33] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” in *Proc. ICLR*, 2015.

[34] H.-S. Heo, K. Nam, B.-J. Lee, Y. Kwon, M. Lee, Y. J. Kim, and J. S. Chung, “Rethinking session variability: Leveraging session embeddings for session robustness in speaker verification,” in *Proc. ICASSP*, 2023.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Proc. NeurIPS*, vol. 32, 2019.

[36] D. P. Kingma, J. Ba, Y. Bengio, and Y. LeCun, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015.

[37] S. Wang, Y. Yang, Y. Qian, and K. Yu, “Revisiting the statistics pooling layer in deep speaker embedding learning,” in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.

[38] O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, “The 2018 nist speaker recognition evaluation,” in *Proc. Interspeech*, 2019.