HYU submission for the SASV challenge 2022: Reforming speaker embeddings with spoofing-aware conditioning

Jeong-Hwan Choi, Joon-Young Yang*, Ye Rin Jeoung, and Joon-Hyuk Chang*

Department of Electronic Engineering
Hanyang University, Seoul, Republic of Korea
{brent1104, dreadbird, jyr0328, jchang}@hanyang.ac.kr

Abstract

In this paper, we introduce the Hanyang University spoofing-aware speaker verification (SASV) system submitted for the SASV challenge 2022. Our strategy is to learn spoofing-aware speaker embeddings (SASEs) that can effectively produce SASV scores with a simple cosine similarity scoring backend. To achieve this, we build a neural-network-based SASE model that uses a spoofing countermeasure (CM) embedding and speaker embedding to produce an SASE. The baseline anti-spoofing model is used to extract CM embeddings, and ResNet-34- and Res2Net-based models are employed for the speaker embedding extraction. When evaluated on the ASVspoof2019 logical access dataset, our best tandem (i. e., the cascade of anti-spoofing and speaker verification) and proposed SASV systems achieved SASV equal error rates of 0.1924% and 0.1817% on the developement set and 0.3911% and 0.2793% on the evaluation set partitions, respectively.

Index Terms: speaker verification, anti-spoofing, spoofing-aware speaker verification

1. Introduction

Spoofing countermeasures (CMs) for detecting synthesized, machine-generated speech utterances are an essential requirement for the design of reliable speaker verification (SV) systems in practical voice biometrics applications. Several studies have proposed spoofing CMs to detect spoofed speech generated using speech synthesis [1] or voice conversion [2] techniques as well as speech utterances replayed through voice recording devices [3]. Most of these studies employed CM systems as independent preprocessing or postprocessing modules for SV systems.

More recently, joint system-level integration of CM and SV systems have been studied [4–6]. In [4], an i-vector [7] space was explored to model synthesis-channel subspace for voice conversion attacks and jointly perform SV and anti-spoofing. A multitask learning [8] approach was adopted in [5] to learn neural network (NN)-base embeddings containing both speaker and spoofing information. In [6], an NN-based backend was proposed to classify the combined set of CM and SV embeddings, extracted from the test and enrollment utterances, to one of the target, nontarget, and spoofed trial classes.

In line with the abovementioned studies, a series of ASVspoof 2015–2021 challenges [9–12] and spoofing-aware speaker verification (SASV) challenge 2022 [13] provided protocols for the evaluation of SV systems in spoofing attack scenarios. Specifically, unlike the previous ASVspoof challenges [9–12], the SASV challenge [13] provokes a paradigm change from the development of CMs for a fixed SV system toward the joint system-level integration of CM and SV systems.

This paper describes the Hanyang University solution for the SASV challenge 2022 [13]. Our SASV system is designed to have characteristics similar to a cascade of an anti-spoofing and SV system, but aims to learn spoofing-aware speaker embeddings (SASEs) that can effectively produce SASV scores as cosine similarity metrics. The proposed SASE model comprises several NN layers and integrates embeddings extracted from the pretrained anti-spoofing and speaker embedding models. More specifically, the SASE model computes weighted summation of an (unmodified) input speaker embedding and modified speaker embedding by employing the probabilities of input speech being bonafide and spoof as the weighting factors. Herein, the modified speaker embedding is obtained by applying feature-wise linear modulation (FiLM) [14] to an input speaker embedding, where the parameters for the FiLM are estimated using a CM embedding.

The rest of this paper is organized as follows: Sections 2 and 3 describe the proposed SASV system and specifications regarding the system development and evaluation. Section 4 provides the experimental results and analysis, and Section 5 concludes the study.

2. System description

2.1. Anti-spoofing model

In this study, we use the pretrained AASIST [15] anti-spoofing model, provided as part of the baseline system for the SASV challenge 2022 [13], without retraining. The AASIST model uses the first 64,600 audio samples of an utterance as input [15]. Detailed model architecture is described in [15].

2.2. Speaker embedding model

We consider two different speaker embedding model architectures in this study. The first model is ResNet-34 exactly identical to that described in [16]. This model uses 64-dimensional (64D) mel-filterbank energies (MFBEs) as input and applies channel-dependent attentive statistics pooling [17] to the multiple hierarchies of feature maps for the speaker embedding extraction [16]. Further details are described in [16]. The second model is Res2Net, which substitutes the basic building block of the ResNet-34 with the Res2Net module [18], except for the first block, and uses 80D MFBEs as input. The base width, scale, and base channel size parameters [18] are set to 16, 4, and 48, respectively. The ResNet-34 and Res2Net have 13.64 and 12.19 million parameters, respectively.

For both models, 256D speaker embeddings are extracted from the penultimate layer with a batch normalization (BN) [19] layer. Unlike the anti-spoofing model described in Section 2.1,

*Equal contributions

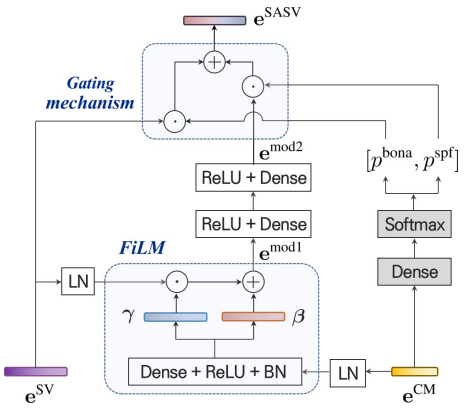

Figure 1: *Block diagram of proposed SASE model.*

full-length utterances are used for the speaker embedding extraction.

2.3. Proposed SASE model

2.3.1. Model architecture

Our strategy is to build an NN-based backend that reforms speaker embeddings using the FiLM [14] technique, a simple conditioning method for NN layers based on affine transformation. Specifically, given a pair of CM and speaker embeddings, a CM embedding is used to calculate the FiLM parameters (i. e., scale and shift) as follows:

$$[\boldsymbol{\gamma}^\top, \boldsymbol{\beta}^\top]^\top = \text{BN}\big(f\big(\mathbf{W}_1^\top \text{LN}\big(\mathbf{e}^{\text{CM}}\big) + \mathbf{b}_1\big)\big) \in \mathbb{R}^{2d_{\text{sv}}}, \quad (1)$$

where $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{d_{\text{sv}}}$ are scale and shift parameters for the FiLM, and $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{cm}} \times 2d_{\text{sv}}}$ and $\mathbf{b}_1 \in \mathbb{R}^{2d_{\text{sv}}}$ are trainable weight and bias, respectively. $\mathbf{e}^{\text{CM}}$ denotes a CM embedding, and d_{sv} and d_{cm} denote the dimensions of speaker and CM embeddings, respectively. LN$(\cdot)$, BN$(\cdot)$ and $f(\cdot)$ denote layer normalization (LN) [20], BN [19] and rectifier [21], respectively. Subsequently, the input speaker embedding is processed through the FiLM layer and two fully connected layers as follows:

$$\mathbf{e}^{\text{mod1}} = \boldsymbol{\gamma} \odot \text{LN}\big(\mathbf{e}^{\text{SV}}\big) + \boldsymbol{\beta} \in \mathbb{R}^{d_{\text{sv}}}, \quad (2)$$

$$\mathbf{e}^{\text{mod2}} = \mathbf{W}_3^\top f\big(\mathbf{W}_2^\top f\big(\mathbf{e}^{\text{mod1}}\big) + \mathbf{b}_2\big) + \mathbf{b}_3 \in \mathbb{R}^{d_{\text{sv}}}, \quad (3)$$

where $\mathbf{e}^{\text{SV}}$ denotes a speaker embedding, $\odot$ denotes elementwise multiplication, and $\mathbf{W}_2, \mathbf{W}_3 \in \mathbb{R}^{d_{\text{sv}} \times d_{\text{sv}}}$ and $\mathbf{b}_2, \mathbf{b}_3 \in \mathbb{R}^{d_{\text{sv}}}$ are trainable weights and biases, respectively. Finally, the reformed speaker embedding, $\mathbf{e}^{\text{SASV}}$, is obtained as the gated summation of the modulated and unmodulated (input) speaker embeddings.

$$\mathbf{e}^{\text{SASV}} = p^{\text{spf}}\mathbf{e}^{\text{mod2}} + p^{\text{bona}}\mathbf{e}^{\text{SV}}, \quad (4)$$

where p^{bona} and p^{spf} denote the probabilities of input speech being bonafide and spoofed, respectively. Herein, p^{bona} and p^{spf} are obtained from the pretrained AASIST [15] model. Note that as expressed in Eq. (4), the proposed SASE model was designed to maintain the input speaker embedding if bonafide speech is observed ($p_{\text{bona}} \simeq 1$), but reform it through CM-conditioned FiLM if spoofed speech is observed ($p_{\text{spf}} \simeq 1$). The block diagram of the SASE model is shown in Fig. 1. The shaded blocks in Fig. 1 denote the output layer of the pretrained AASIST model and are frozen during the training of the proposed SASE model.

Given a trial, the SASV score is calculated as the cosine similarity between the pair of reformed speaker embeddings.

Table 1: *Statistics of ASVspoof2019 LA [11] dataset*

Partition	#speakers		#utterances		Attack types
	Male	Female	Bonafide	Spoofed	
Train	8	12	2,580	22,800	A01–A06
Dev	4	6	2,548	22,296	A01–A06
Eval	21	27	7,355	63,882	A07–A19

Table 2: *Description of SV-EER, SPF-EER, and SASV-EER*

Metric (Abbreviation)	Target	Nontarget	Spoof
SV-EER (EER_{SV})	+	-	
SPF-EER (EER_{SPF})	+		-
SASV-EER (EER_{SASV})	+	-	-

2.3.2. Loss function

To train the proposed SASE model, a minibatch is composed as described in the following. Given S speakers in the training set, we first sample s unique speakers and randomly select m, n, and r utterances for each speaker. These utterances are used as bonafide enrollment, bonafide test, and spoofed test utterances, respectively. Subsequently, all the selected utterances are processed through Eqs. (1) – (4), and two score matrices, $\mathbf{A}_{\text{bona}} \in \mathbb{R}^{ms \times ns}$ and $\mathbf{A}_{\text{spf}} \in \mathbb{R}^{ms \times rs}$, are calculated to consider all possible combinations of target, nontarget, and spoofed trials for the selected ms enrollment utterances. The elements of $\mathbf{A}_{\text{bona}}$ and $\mathbf{A}_{\text{spf}}$ are cosine similarities between pairs of SASV embeddings. Finally, the loss function for training the proposed SASE model is defined as follows:

$$a'_{ij} = \sigma(w \cdot a_{ij} + b), \quad (5)$$

$$L = \sum_{\substack{0 \le i < ms, \\ 0 \le j < (n+r)s}} -\frac{t_{ij} \ln(a'_{ij}) + (1 - t_{ij}) \ln(1 - a'_{ij})}{m(n+r)s^2}, \quad (6)$$

where i and j denote row and column indices of a concatenated score matrix $\mathbf{A} = [\mathbf{A}_{\text{bona}}\ \mathbf{A}_{\text{spf}}] \in \mathbb{R}^{ms \times (n+r)s}$, and a_{ij} denotes an element of $\mathbf{A}$. w and b are trainable scale and shift parameters [22], and $\sigma(\cdot)$ denotes a sigmoid function. t_{ij} denotes the binary target labels for a_{ij}, whose value is set to 1 for a target trial and 0 otherwise.

3. Experimental setup

3.1. Training specifications

The SASV challenge 2022 allowed the use of the "train" and "dev" partitions of the ASVspoof2019 logical access (LA) [11] dataset as well as the VoxCeleb2 [23] dataset for the system development. The use of any non-speech audio data was also allowed for data augmentation purposes.

The speaker embedding models described in Section 2.2 were trained using the "dev" partition of the VoxCeleb2 [23] dataset. Data augmentation was conducted using the MUSAN [24] babble and music samples, noise samples from the deep noise suppression challenge 2020 [25], and simulated room impulse responses [26], creating 4 additional copies of the original audio samples. Note that the augmented training samples for the ResNet-34 and the Res2Net models were created using different random seeds. Further details regarding the model training procedure are described in [16].

Table 3: *SASV performance of speaker embedding models and AASIST [15] anti-spoofing model*

Model	$\text{EER}_{\text{SV}}(\%)$	$\text{EER}_{\text{SPF}}(\%)$	$\text{EER}_{\text{SASV}}(\%)$
	dev / eval	dev / eval	dev / eval
AASIST	46.3 / 48.9	**0.07 / 0.66**	15.9 / 24.5
ECAPA-TDNN	1.27 / 0.84	19.0 / 29.3	16.2 / 22.4
ResNet-34	0.54 / 0.43	15.3 / 25.4	12.5 / 19.2
Res2Net	**0.20 / 0.20**	**14.8 / 24.7**	**12.1 / 18.7**

The AASIST anti-spoofing model was trained using the "train" partition of the ASVspoof2019 LA [11] dataset, whose statistics are summarized in Table 1. For further details regarding the training procedure, please refer to [15].

The proposed SASE model was trained using the same "train" partition of the ASVspoof2019 LA [11] dataset. We set $s = 20$, $m = 1$, $n = 1$, and $r = 4$ in Eq. (6), and initialized w and b in Eq. (5) to 15 and -5, respectively. A single training epoch was defined as iterations over 200 minibatches, and the training was conducted for 50 epochs. Nadam [27] optimizer was used for the training with an initial learning rate of 0.00008 and a momentum decay of 0.004, and $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ in Eqs. (1) and (3) were ℓ_2-regularized with a scale of 0.00005. The input 160D CM embeddings were extracted from the penultimate layer of the AASIST [15] model.

3.2. Evaluation

Three evaluation metrics were employed to measure SV, anti-spoofing, and SASV performances in terms of the equal error rate (EER) [13]. These metrics are denoted as SV-EER, SPF-EER, and SASV-EER, respectively, and differ in the definition of negative classes. Table 2 summarizes the types of positive and negative classes considered for the metrics computation[1].

The evaluation was conducted using the "eval" partition trials of the ASVspoof2019 LA [11] dataset, which comprised 5,370 target, 33,327 nontarget, and 63,882 spoofed trials. The "dev" partition trials comprised 1,484 target, 5,768 nontarget, and 22,296 spoofed trials [11], and was used to select the model exhibiting the lowest SASV-EER. Detailed statistics regarding the "dev" and "eval" partitions are summarized in Table 1.

For comparison purposes, we also employed the baseline ECAPA-TDNN [17] speaker embedding model to train the proposed SASE model, with full-length utterances as input. Moreover, the tandem SASV systems were implemented, for which the SASV score was set identical to the SV score if the CM score was above a predefined CM threshold, but to -1 otherwise. The CM score and SV score were calculated as described in the following. Given a trial, the SV score was calculated as the cosine similarity between the speaker embeddings extracted from the test and enrollment utterances, and the CM score was obtained as the probability of the test utterance being bonafide, p^{bona}. The CM thresholds for the tandem systems were determined using the "dev" partition trials under the lowest SASV-EER criterion. Consequently, we obtained the CM thresholds of 0.9135, 0.908, and 0.96 for the ECAPA-TDNN-, ResNet-34- and Res2Net-based tandem systems, respectively. Note that the tandem systems built with the different speaker embedding models had different optimal CM thresholds because the SASV performance depended on the SV scores assigned to the false positives (i. e., falsely detected spoofed trials).

[1]Python 3.9.5, SciPy 1.7.3, and sklearn 1.0.2 were used.

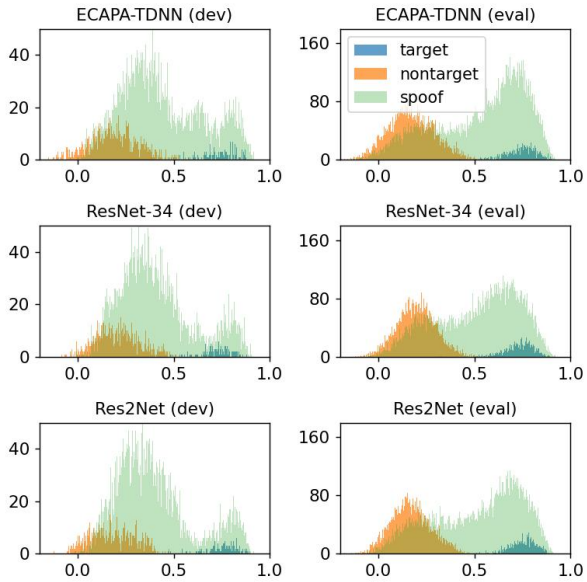

Figure 2: *Histograms of SV scores.*

4. Experimental results and analysis

4.1. Standalone CM and SV systems

Table 3 summarizes the SASV performance of the CM system based on the AASIST [15] anti-spoofing model and SV systems built with the three different speaker embedding models. Because the CM score represented the probability of a test utterance being bonafide, but did not consider the relationship between test and enrollment utterances, the SV-EERs of the AASIST model were close to 50%. Comparing the SV performances of the three speaker embedding models, both our ResNet-34 and Res2Net were significantly superior to the baseline ECAPA-TDNN model. Specifically, the Res2Net outperformed the others by considerable margins on both "dev" and "eval" partitions.

Fig. 2 shows the histograms of the SV scores obtained from the three different SV systems. First, for both ResNet-34 and Res2Net models, the SV scores for the spoofed trials measured on the "dev" partition seemed approximately bi-modal and were more heavily distributed toward the nontarget scores side than the target scores side. This indicates that although the speaker embedding models were not trained to discriminate spoofed speech, their speaker embeddings could help reject spoofed trials of target speakers to an extent. Indeed, the ECAPA-TDNN-, ResNet-34-, and Res2Net-based SV systems achieved SPF-EERs of 19.0%, 15.3%, and 14.8%, respectively, as presented in Table 3. Second, on the same "dev" partition, the overlap between the scores for the spoofed trials and those for the target trials was apparently larger in the ECAPA-TDNN model than in the ResNet-34 and Res2Net. This supports the SPF-EERs of the ResNet-34- and Res2Net-based SV systems being significantly lower than the SPF-EER of the ECAPA-TDNN-based SV system. Finally, on the "eval" partition, the scores for the spoofed trials were generally more heavily distributed toward the target scores side, unlike the trends observed on the "dev" partition. This suggests that the types of spoofing attacks included in the "eval" partition are more challenging to reject compared to those included in the "dev" partition.

Table 4: *SASV performance of tandem systems*

Speaker embedding	EER_{SV}(%) dev / eval	EER_{SPF}(%) dev / eval	EER_{SASV}(%) dev / eval
ECAPA-TDNN	1.28 / 0.92	**0.07** / 0.62	0.61 / 0.80
ResNet-34	0.54 / 0.54	**0.07** / 0.59	0.34 / 0.56
Res2Net	**0.28 / 0.43**	0.13 / **0.37**	**0.19 / 0.39**

Table 5: *SASV performance of proposed systems*

Speaker embedding	EER_{SV}(%) dev / eval	EER_{SPF}(%) dev / eval	EER_{SASV}(%) dev / eval
ECAPA-TDNN	1.30 / 0.91	**0.07** / 0.36	0.61 / 0.69
ResNet-34	0.62 / 0.54	**0.07** / 0.38	0.34 / 0.47
Res2Net	**0.28 / 0.28**	**0.07 / 0.28**	**0.18 / 0.28**

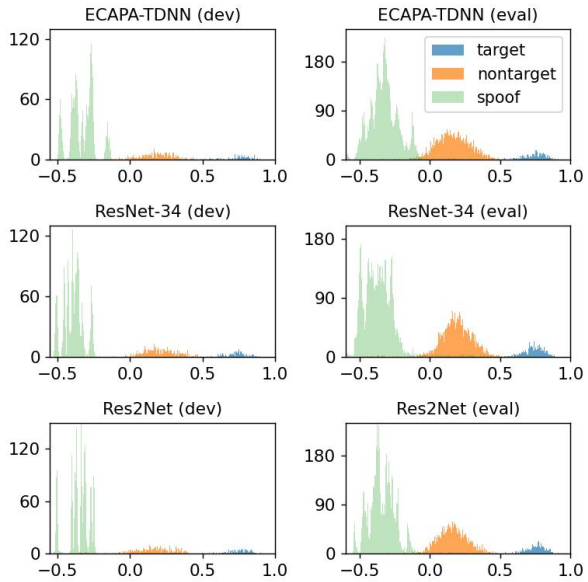

Figure 3: *Histograms of SASV scores.*

4.2. Tandem and proposed SASV systems

Tables 4 and 5 summarize the SASV performances of the tandem and proposed SASV systems. The SASV-EERs of the proposed systems were generally improved on the "eval" partition compared to the tandem systems, but maintained on the "dev" partition. Apparently, this was attributed to the decrease in the SPF-EERs, as the SV-EERs were generally maintained or slightly increased, except for some cases. These trends indicate that the proposed SASE model operated as intended, because ideally, the SASV embedding should be identical to the speaker embedding for a bonafide trial. Indeed, we confirmed that the SASV scores for the bonafide trials, correctly detected by the CM system, were almost the same for both tandem and proposed systems. Thus, it can be inferred that the proposed SASE model successfully learned to produce embeddings better capable of dealing with bonafide and spoofed trials than the tandem systems. Note that both false negatives and false positives caused by the tandem CM systems were corrected by the proposed method on the "eval" partition, but not in the "dev" partition, probably because there was small room for

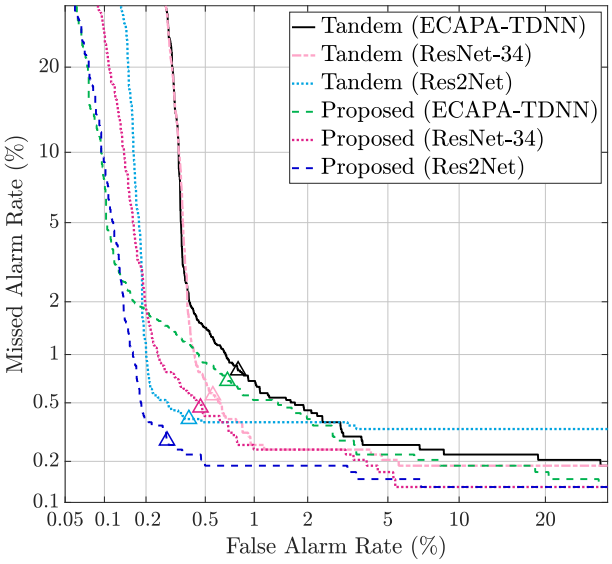

Figure 4: *DET curves of tandem and proposed SASV systems on evaluation set. "△" denotes operating points for SASV-EERs.*

improvement (i. e., SPF-EERs of 0.07% were small enough). The histograms of the SASV scores obtained from the proposed systems are shown in Fig. 3. These historgrams show that the learned SASEs produced SASV scores significantly smaller for the spoofed trials than for the nontarget trials. Finally, we found that the changes in SV-EERs (from 0.54 to 0.62 for the ResNet-34-based systems on the "dev" and 0.43 to 0.28 for the Res2Net-based systems on the "eval" partitions) were attributed to the difference in the thresholds at which the SASV-EERs were measured. For example, the thresholds were determined to 0.441 and 0.454 for the Res2Net-based tandem and proposed SASV systems, respectively, and thus, some nontarget trials, whose SASV scores were between 0.441 and 0.454, were corrected as true negatives. The opposite was observed between the ResNet-34-based tandem and proposed systems. Fig. 4 shows the detection error tradeoff (DET) curves on the "eval" partition, which demonstrate improved operating characteristics of the proposed systems over the tandem systems.

5. Conclusions

In this study, we proposed an NN-based SASV system to improve a tandem SASV system by maintaining SV performance for the correctly detected bonafide trials, while improving erroneous hard decisions of the tandem CM system. We submitted the results obtained from the proposed SASV system built with our Res2Net-based speaker embedding model, which exhibited the lowest SASV-EER on the "dev" partition among the considered systems. The experimental results demonstrated the effectiveness of the proposed method, suggesting that the joint integration of CM and speaker embeddings is a promising direction for SASV.

6. References

[1] P. L. De Leon, M. Pucher, and J. Yamagishi, "Evaluation of the vulnerability of speaker verification to synthetic speech," in *Proc. Odyssey*, 2010, pp. 151–158.

[2] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in

Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2012, pp. 4401–4404.

[3] J. Villalba and E. Lleida, "Preventing replay attacks on speaker verification systems," in *Proc. IEEE Int. Carnahan Conf. Security Tech.*, 2011, pp. 1–8.

[4] A. Sizov, E. Khoury, T. Kinnunen, Z. Wu, and S. Marcel, "Joint speaker verification and antispoofing in the i-vector space," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 821–832, 2015.

[5] J. Li, M. Sun, X. Zhang, and Y. Wang, "Joint decision of anti-spoofing and automatic speaker verification by multi-task learning with contrastive loss," *IEEE Access*, vol. 8, pp. 7907–7915, 2020.

[6] A. Gomez-Alanis, J. A. Gonzalez-Lopez, S. P. Dubagunta, A. M. Peinado, and M. M. Doss, "On joint optimization of automatic speaker verification and anti-spoofing in the embedding space," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1579–1593, 2020.

[7] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, 2010.

[8] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[9] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. INTERSPEECH*, 2015, pp. 2037–2041.

[10] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in *Proc. INTERSPEECH*, 2017, pp. 2–6.

[11] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv:1904.05441*, 2019.

[12] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, and H. Delgado, "ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection," *arXiv:2109.00537*, 2021.

[13] J.-W. Jung, H. Tak, H.-J. Shim, H.-S. Heo, B.-J. Lee, S.-W. Chung, H.-G. Kang, H.-J. Yu, N. Evans, and T. Kinnunen, "SASV challenge 2022: A spoofing-aware speaker verificiation challenge evaluation plan," *arXiv:2201.10283*, 2022.

[14] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "FiLM: Visual reasoning with a general conditioning layer," *arXiv:1709.07871*, 2017.

[15] J.-W. Jung, H.-S. Heo, H. Tak, H.-J. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," *arXiv:2110.01200*, 2021.

[16] J.-Y. Yang and J.-H. Chang, "Task-specific optimization of virtual channel linear prediction-based speech dereverberation front-end for robust far-field speaker verification," *arXiv:2112.13569*, 2021.

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

[18] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, 2019.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv:1607.06450*, 2016.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machine," in *Proc. Int. Conf. Mach. Learn.*, 2010.

[22] 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.

[23] A. N. Chung, Joon Son and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. INTERSPEECH*, 2018, pp. 1086–1090.

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

[25] C. K. A. Reddy *et al.*, "The INTERSPEECH 2020 deep noise suppression challenge: Datasets, subjective speech quality and testing framework," *arXiv:2001.08662*, 2020.

[26] 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. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5220–5224.

[27] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–4.