



# Towards single integrated spoofing-aware speaker verification embeddings

Sung Hwan Mun<sup>1,\*</sup>, Hye-jin Shim<sup>2,\*</sup>, Hemlata Tak<sup>3,\*</sup>, Xin Wang<sup>4</sup>, Xuechen Liu<sup>2,5</sup>, Md Sahidullah<sup>6</sup>, Myeonghun Jeong<sup>1</sup>, Min Hyun Han<sup>1</sup>, Massimiliano Todisco<sup>3</sup>, Kong Aik Lee<sup>7</sup>, Junichi Yamagishi<sup>4</sup>, Nicholas Evans<sup>3</sup>, Tomi Kinnunen<sup>2</sup>, Nam Soo Kim<sup>1</sup>, and Jee-weon Jung<sup>8,†</sup>

<sup>1</sup>Seoul National University, South Korea, <sup>2</sup>University of Eastern Finland, Finland, <sup>3</sup>EURECOM, France, <sup>4</sup>National Institute of Informatics, Japan, <sup>5</sup>Inria, France, <sup>6</sup>TCG CREST, India, <sup>7</sup>Institute for Infocomm Research, A\*STAR, Singapore, <sup>8</sup>Carnegie Mellon University, USA  
shmun@hi.snu.ac.kr, hyejin.shim@uef.fi, tak@eurecom.fr, jeeveonj@andrew.cmu.edu

## Abstract

This study aims to develop a single integrated spoofing-aware speaker verification (SASV) embeddings that satisfy two aspects. First, rejecting non-target speakers' input as well as target speakers' spoofed inputs should be addressed. Second, competitive performance should be demonstrated compared to the fusion of automatic speaker verification (ASV) and countermeasure (CM) embeddings, which outperformed single embedding solutions by a large margin in the SASV2022 challenge. We analyze that the inferior performance of single SASV embeddings comes from *insufficient amount of training data and distinct nature of ASV and CM tasks*. To this end, we propose a novel framework that includes multi-stage training and a combination of loss functions. Copy synthesis, combined with several vocoders, is also exploited to address the lack of spoofed data. Experimental results show dramatic improvements, achieving an SASV-EER of 1.06% on the evaluation protocol of the SASV2022 challenge.

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

## 1. Introduction

While today's state-of-the-art automatic speaker verification (ASV) systems [1, 2] offer exceptional reliability across diverse scenarios and domains [3, 4], their vulnerabilities to spoofing and deepfake attacks are well acknowledged. Such malicious attacks can be generated using text-to-speech (TTS) and voice conversion (VC) algorithms, for instance, in order to deceive an ASV system and provoke an increased rate of false positives. As a result, the development of spoofing and deepfake detection systems, known as countermeasures (CMs), has attracted considerable research interest in recent years [5–7].

The majority of CM solutions take the form of binary classifiers. Their role is to protect the ASV system by determining whether a given utterance is bona fide (i.e., genuine) or spoofed. While the development of such separate CM and ASV sub-systems is common to the majority of related research, the recent Spoofing-Aware Speaker Verification (SASV) initiative [8] was formed in 2022 to promote the exploration of jointly optimized CM and ASV solutions as well as integrated approaches whereby the same tasks are performed by a single classifier.

The initiative was well supported and attracted the registration of 53 participants. While a majority of participants utilized fusion-based approaches, encouragingly, some participants explored end-to-end (E2E) approaches and the training of single, integrated systems [9–11]. E2E approaches involve the training

of an SASV model to extract SASV embeddings directly (similar to speaker embeddings for an ASV task) without optimization of separate ASV and CM models. These, however, significantly under-perform solutions based upon the fusion of independent, pre-trained CM and ASV sub-systems at embedding, score, or decision levels [12–17].

Even if their performance is currently behind that of their more traditional counterparts, single, integrated systems are not without appeal. A single classifier might be more efficient than two, depending on their relative complexity. Both CMs and ASV subsystems have potential to deflect spoofing attacks [18], and jointly-optimized solutions are gaining ground on separate, independently optimized alternatives [14, 19–21]. Single classifier solutions might also help to avoid the more costly joint optimization or adaptation of dual classifier solutions which might be applied periodically to protect reliability in the face of newly emerging threats. Finally, given that they have received far less research attention than their dual classifier counterparts, it is sensible to assume that there might be as-yet untapped potential to improve the competitiveness of single classifier solutions with further research investment. We believe that single, integrated solutions still merit attention.

To this end, we propose a novel multi-stage framework for training single SASV embedding extractors. The framework employs different stages to leverage vast amounts of data, including VoxCeleb2 [22] as well as copy synthesis (CS) data [23]. Furthermore, considering the characteristics of SASV task, involving two different tasks in nature, delicate optimization using diverse combinations of loss functions is explored to go beyond the simple binary classification. We conducted extensive experiments using two recent speaker embedding extractors, namely a selective kernel attention-based time delay neural network (SKA-TDNN) [24] and a multi-scale feature aggregation conformer (MFA-Conformer) [3]. Together, these contributions narrow significantly the performance gap between embedding-level, fusion-based and single, integrated SASV solutions.<sup>1</sup>

## 2. Proposed SASV framework

Previous studies in SASV have shown the improvements to performance that can be found from using large-scale datasets [8, 12, 16, 17, 20]. However, while bona fide data is available in abundance, spoofed data is comparatively scarce. New techniques are needed to compensate for this data imbalance while still exploiting the volume of bona fide data needed to support reliable ASV. We propose a multi-stage training scheme for the extraction of SASV embeddings and the use of appropriate diverse loss functions.

\*Equal contribution.

† Work done while author was in Naver Corporation, South Korea.

<sup>1</sup>Code and models are available in <https://github.com/sasv-challenge/ASVspoof5-SASVBaseline>.

First, the ability to discriminate between target and bona fide non-target speakers can be learned using the VoxCeleb2 database which contains data collected from thousands of bona fide speakers. Second, we augment the model with the ability to discriminate between bona fide and spoofed inputs by using large-scale data generated through an oracle speech synthesis system, referred to as copy synthesis [25]. Third, we fine-tune the model using the ASVspoof 2019 Logical Access (LA) database which contains a mix of bona fide and spoofed utterances. Figure 1 illustrates the multi-training scheme.

## 2.1. Stage 1: Speaker classification-based pre-training

In Stage 1, pre-training is performed using VoxCeleb2 so that the model learns to distinguish between target and non-target trials, both of which are bona fide (i.e., conventional ASV). We optimize the model using the equal-weighted summation of additive angular margin (AAM) softmax loss [26] and contrastive loss functions. The combination of two functions has synergistic effects, resulting in better performance than when using either alone [24, 27, 28].

Given a mini-batch of SASV embeddings,  $\mathbf{x}_{i,1}$  and  $\mathbf{x}_{i,2}$ , where  $\mathbf{x}_{i,k}$  denotes a SASV embedding extracted from the  $k^{\text{th}} \in [1, 2]$  utterance of the  $i^{\text{th}} \in [1, \dots, C_{\text{spk}}]$  speaker, the contrastive loss function is defined as follows:

$$\mathcal{L}_{\text{cont}}^{\text{asv}} = -\frac{1}{N} \sum_i \log \frac{\exp(f(\mathbf{x}_{i,1}, \mathbf{x}_{i,2}))}{\sum_j \exp(f(\mathbf{x}_{i,1}, \mathbf{x}_{j,2}))}, \quad (1)$$

where  $N$  is the number of speakers within a mini-batch and  $f(\cdot) = \alpha \cos(\cdot) + \beta$  is a cosine similarity function between two vectors with trainable scale  $\alpha$  and bias  $\beta$  parameters. Given a target speaker label  $\mathbf{y}_{i,k} = i \in [1, \dots, C_{\text{spk}}]$  corresponding to  $\mathbf{x}_{i,k}$ , the AAM softmax loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{aam}}^{\text{asv}} &= \text{AAMsoftmax}([\mathbf{x}_{i,k}, \mathbf{y}_{i,k}]_{\forall(i,k)}; C_{\text{spk}}) \\ &= -\frac{1}{2N} \sum_{i,k} \log \frac{e^{s \cos(\theta_{\mathbf{y}_{i,k}, i+m})}}{e^{s \cos(\theta_{\mathbf{y}_{i,k}, i+m})} + \sum_{j \neq \mathbf{y}_{i,k}^c} e^{s \cos(\theta_{j,i})}}, \end{aligned} \quad (2)$$

where  $s$  is a scale factor,  $m$  is a margin,  $\cos(\theta_{j,i})$  is the normalized dot product between the weight for the  $j^{\text{th}} \in [1, \dots, C_{\text{spk}}]$  class and the input SASV embedding  $\mathbf{x}_{i,k}$ . Finally, the ASV loss function for Stage 1 is defined as  $\mathcal{L}^{\text{asv}} = \mathcal{L}_{\text{cont}}^{\text{asv}} + \mathcal{L}_{\text{aam}}^{\text{asv}}$ .

## 2.2. Stage 2: copy synthesis training with adapted SASV loss functions

The training of CMs also requires large databases of both spoofed and bona fide data. Compared to abundant bona fide data, generating spoofed data remains technically demanding and time-consuming.<sup>2</sup> Nevertheless, generating spoofed data with a wide variety of different algorithms is essential for generalization, namely the reliable detection performance in the face of unseen spoofing attacks. Greater quantities, more on par with that of bona fide data, may also help to avoid the impact of data imbalance. Moreover, no systematic study has investigated the loss function which is most appropriate for SASV, even though the training objective is critical. To this end, we use copy synthesis (CS) as a data augmentation to enlarge the volume of spoofed data and compare different loss functions.

<sup>2</sup>To give an idea of the scale, the creation of the ASVspoof 2019 dataset [6] required more than 6 months of intensive effort from several contributors.

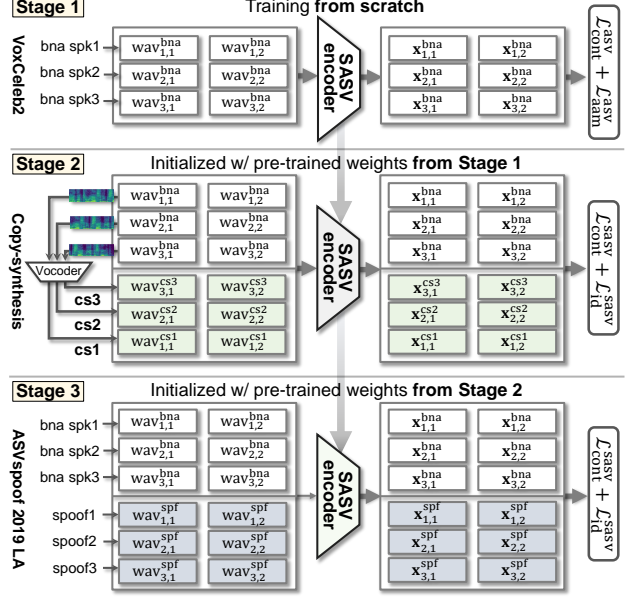


Figure 1: Proposed SASV framework with three training Stages and diverse loss functions.  $wav^{bna}$  and  $wav^{spf}$  denote bona fide and spoof waveform of  $i$ -th speaker's  $j$ -th utterance.  $CS_k$  denotes the copy synthesis spoofed data from  $k$ -th speaker.

**Utilization of copy synthesis data.** Recently, an efficient means to generate massive databases for spoofing CM training was proposed in [23]. It uses neural network-based vocoders and a CS technique to generate spoofs from bona fide utterances using oracle acoustic features (e.g. F0 and mel spectrogram representations), and speaker embeddings (e.g. x-vectors) [29, 30]. Similar ideas have been reported previously [29–34], even though most of the previous studies use traditional non-neural vocoders. Adopting the same idea, we use the data generated via CS from VoxCeleb2 and ASVspoof 2019 LA training datasets. We employ the same four neural vocoders as described in [23]: HiFiGAN [35]; WaveGlow [36]; a neural source-filter model (NSF) [37]; a NSF model with a HiFiGAN discriminator. This set of vocoders includes flow, generative adversarial network (GAN), and differentiable digital signal processing techniques [38], and can generate waveforms at real-time speed, making them ideally suited to CS using large datasets.

**Adaptive contrastive loss for SASV.** The contrastive loss function, which is widely used in speaker verification [39] and also in Stage 1, maximizes speaker discrimination in the embeddings corresponding to different speakers. However, in the case of a spoof-aware scenario, this may affect (degrade) performance since CM labels (i.e., spoof, target bona fide) are not taken into account. Therefore we explore a contrastive loss function for the training of a single, integrated SASV model. Given a mini-batch of SASV embeddings,  $\mathbf{x}_{i,1}^c$  and  $\mathbf{x}_{i,2}^c$ , where  $\mathbf{x}_{i,k}^c$  denotes a SASV embedding extracted from the  $k^{\text{th}} \in [1, 2]$  utterance of the  $i^{\text{th}} \in [1, \dots, C_{\text{spk}}]$  speaker with CM label  $c \in [\text{bna}, \text{spf}]^3$ , the SASV contrastive loss is defined as follows:

$$\mathcal{L}_{\text{cont}}^{\text{sasv}} = -\frac{1}{N_{\text{bna}}} \sum_i \log \frac{\exp(f(\mathbf{x}_{i,1}^{\text{bna}}, \mathbf{x}_{i,2}^{\text{bna}}))}{\sum_{j,c} \exp(f(\mathbf{x}_{i,1}^{\text{bna}}, \mathbf{x}_{j,2}^c))}, \quad (4)$$

where  $N_{\text{bna}}$  is the number of bona fide speakers within a mini-

<sup>3</sup>The superscript or subscript ‘bna’ and ‘spf’ denote bona fide and spoof data, respectively.

batch and where  $N = N_{\text{bna}} + N_{\text{spf}}$  indicates the total number of bona fide and spoof speakers within a mini-batch. To focus on the differences in artefacts stemming from vocoding, as opposed to other sources of variation (e.g., speakers, phonetic characteristics, etc.), we include both bona fide utterances and their copy synthesized counterparts when creating mini-batches. Given a bona fide utterance,  $\mathbf{x}_{i,k}^{\text{bna}}$ , the corresponding CS utterance is denoted by  $\mathbf{x}_{i,k}^{\text{cs}}$ . Mini-batches are configured with a size of  $4N_{\text{spk}}$  as follows:  $\{[(\mathbf{x}_{i,k}^{\text{bna}}, \mathbf{x}_{i,k}^{\text{cs}})]_{i,k} | \forall i \in [1, \dots, N_{\text{spk}}], \forall k \in [1, 2]\}$ , where the vocoder type used for copy synthesis is randomly selected from the four.

**Integrated & Multi-task SASV identification loss.** The SASV task can be considered as a multi-task problem in which there is a need to distinguish between target and non-target utterances (the traditional task of ASV), as well as bona fide and spoofed utterances (the task of CMs). We investigate two alternative approaches: (1) integrated SASV identification (i.e.,  $C_{\text{spk}} + 1$  class classification) with an additional spoof class; (2) multi-task SASV identification (i.e.,  $C_{\text{spk}}$  speaker classification and binary spoofing detection tasks).

First, bona fide and spoof SASV embeddings are defined by  $\mathbf{x}_{i,k}^{\text{bna}}$  and  $\mathbf{x}_{i,k}^{\text{spf}}$ . We set their corresponding SASV identification targets to  $\hat{\mathbf{y}}_{i,k}^{\text{bna}} = i \in [1, \dots, C_{\text{spk}}]$  and  $\hat{\mathbf{y}}_{i,k}^{\text{spf}} = C_{\text{spk}} + 1$ , respectively. We use the AAM softmax [26] loss to perform  $C_{\text{spk}} + 1$  class identification. For given the input SASV embeddings  $\mathbf{x}_{i,k}^c$  and their corresponding SASV identification targets  $\hat{\mathbf{y}}_{i,k}^c$ , the AAM softmax is defined as:

$$\mathcal{L}_{\text{id}_1}^{\text{sasv}} = \text{AAMsoftmax}([\mathbf{x}_{i,k}^c, \hat{\mathbf{y}}_{i,k}^c]_{\forall(i,k,c)}; C_{\text{spk}} + 1). \quad (5)$$

Second, instead of learning the SASV embeddings from an integrated single output logit, a multi-task learning [40] framework can be utilized. Speaker and CM labels are denoted by  $\mathbf{y}_{i,k}^c = i \in [1, \dots, C_{\text{spk}}]$  and  $\tilde{\mathbf{y}}_{i,k}^c = c \in [\text{bna}, \text{spf}]$  for SASV embedding  $\mathbf{x}_{i,k}^c$ , respectively. The multi-task SASV identification loss is then given by:

$$\mathcal{L}_{\text{id}_2}^{\text{sv}} = \text{AAMsoftmax}([\mathbf{x}_{i,k}^c, \mathbf{y}_{i,k}^c]_{\forall(i,k,c)}; C_{\text{spk}}), \quad (6)$$

$$\mathcal{L}_{\text{id}_2}^{\text{spf}} = \text{AAMsoftmax}([\mathbf{x}_{i,k}^c, \tilde{\mathbf{y}}_{i,k}^c]_{\forall(i,k,c)}; 2), \quad (7)$$

$$\mathcal{L}_{\text{id}_2}^{\text{sasv}} = \mathcal{L}_{\text{id}_2}^{\text{sv}} + \mathcal{L}_{\text{id}_2}^{\text{spf}}, \quad (8)$$

Therefore, the final loss functions in Stage 2 can be either of  $\mathcal{L} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{id}_1}^{\text{sasv}}$  or  $\mathcal{L} = \mathcal{L}_{\text{cont}} + \mathcal{L}_{\text{id}_2}^{\text{sasv}}$ . We compare the two empirically and also explore model training with only  $\mathcal{L}_{\text{cont}}^{\text{sasv}}$ ,  $\mathcal{L}_{\text{id}_1}^{\text{sasv}}$ , or  $\mathcal{L}_{\text{id}_2}^{\text{sasv}}$ .

### 2.3. Stage 3: in-domain fine-tuning

Even though training in Stages 1 and 2 learn to discriminate bona fide non-target and spoof non-target inputs, there is a remaining domain mismatch with the evaluation protocol. Furthermore, artefacts from the acoustic model have yet to be learned. Hence, as a final step, we fine-tune the model using in-domain bona fide and spoofed data contained within the ASVspoof 2019 LA training partition. While training data corresponds to the same domain as that used for evaluation, there are differences in the spoofing algorithms used in their generation. In Stage 3, the same loss functions of Stage 2 are used, but with different mini-batch formulation. Specifically, and different to Stage 2, in Stage 3 we randomly sample  $2N_{\text{spk}}$  bona fide (2 utterances each from  $N_{\text{spk}}$  speakers) and  $2N_{\text{spf}}$  spoofed utterances from the ASVspoof 2019 LA [6] training partition.

Table 1: *Proposed multi-Stage training. Results are reported in terms of SASV-EER (%). trn and trn&dev refer to using train only and both train and dev set of the ASVspoof 2019 LA in Stage 3. Vox<sub>bna</sub>, Vox<sub>bna+cs</sub>, ASV<sub>o</sub>, ASV<sub>bna+cs</sub> indicate VoxCeleb2 (bona fide only), VoxCeleb2 and its CS version, original ASVspoof 2019 LA, and ASVspoof 2019 bona fide and its CS version, respectively. Rows 2, 4, and 6 do not apply Stage 3, and hence only the result after the last applied Stage is reported.*

rows	Stage 1	Stage 2	Stage 3	SKA-TDNN		MFA-Conformer	
	pre-train	CS	fine-tune	trn	trn&dev	trn	trn&dev
1			ASV <sub>o</sub>	10.04	5.94	11.47	7.67
2	Vox <sub>bna</sub>			16.74		20.22	
3	Vox <sub>bna</sub>		ASV <sub>o</sub>	2.67	<b>1.25</b>	2.13	1.49
4		Vox <sub>bna+cs</sub>		13.11		14.27	
5		Vox <sub>bna+cs</sub>	ASV <sub>o</sub>	2.49	1.93	1.91	1.35
6	Vox <sub>bna</sub>	Vox <sub>bna+cs</sub>		10.24		12.33	
7	Vox <sub>bna</sub>	Vox <sub>bna+cs</sub>	ASV <sub>o</sub>	<b>1.83</b>	1.56	<b>1.19</b>	<b>1.06</b>
8		ASV <sub>bna+cs</sub>		13.10	10.49	13.68	12.48
9		ASV <sub>bna+cs</sub>	ASV <sub>o</sub>	9.57	6.17	13.46	10.11
10	Vox <sub>bna</sub>	ASV <sub>bna+cs</sub>		5.62	4.93	9.31	8.32
11	Vox <sub>bna</sub>	ASV <sub>bna+cs</sub>	ASV <sub>o</sub>	2.48	1.44	2.72	1.76

## 3. Experimental setup

### 3.1. Datasets and metrics

During the training Stages 1 and 2, we apply CS-based data augmentations using the VoxCeleb2 [22] or/and ASVspoof 2019 LA datasets [6]. During the training Stage 3 and evaluation phases, we only use SASV challenge data to follow the standard protocol [8]. Specifically, in Stage 1 training, we employ the original VoxCeleb2 dataset. In Stage 2, we utilize either the VoxCeleb2 dataset along with the corresponding CS data, or the ASVspoof 2019 LA train and development portions, also paired with the corresponding CS data. Finally, in Stage 3, we solely rely on the ASVspoof 2019 LA train and development partitions, comprising 50,224 utterances (5,128 genuine and 45,096 spoofed) collected from 40 speakers.

Results are reported in terms of three metrics, namely the SASV-EER, the SV-EER, and the SPF-EER [8]. The SASV-EER is computed using target, non-target and spoofed trials, where only target trials should be accepted. The SV-EER is estimated using target and non-target trials. The SPF-EER is computed using target and spoofed trials; it reflects the vulnerability of the ASV system when non-target trials are replaced with spoofed trials.

### 3.2. Implementation details

VoxCeleb2 and ASVspoof 2019 LA utterances are cropped or padded to durations of 2 and 5 seconds respectively. Features are 80-dimensional log mel-filterbank outputs computed with a window size of 25 ms and a frame-shift of 10 ms with mean and variance normalization. All four vocoders described in Section 2.2 were trained from scratch using the VoxCeleb2 development partition and were used to generate copy synthesis data for VoxCeleb2 and the ASVspoof 2019 LA datasets.

For all Stages, we used contrastive and AAM-softmax losses concurrently with AdamW optimizer [41], except for ablation experiments. In Stage 1, we used a batch size of 200 (2 utterances per 100 speakers), a scheduler cycle size of 25 epochs and a maximum learning rate of 0.001 which was decreased by 0.5 between each cycle. For Stage 2, we used a batch size of 200 (2 utterances per 50 speakers for bona fide and corresponding

Table 2: Ablation on the composition of loss functions in Stage 3, after Stage 1 (i.e., row 3 of Table 1). *trn&dev* partition of ASVspoof2019 LA is used for ablation study.

Loss functions	SKA-TDNN			MFA-Conformer		
	SASV	SV	SPF	SASV	SV	SPF
$\mathcal{L}_{\text{cont}}^{\text{sasv}}$	3.62	1.64	4.63	1.89	1.52	2.11
$\mathcal{L}_{\text{id}_1}^{\text{sasv}}$	1.75	2.35	1.06	1.96	2.70	1.01
$\mathcal{L}_{\text{id}_2}^{\text{sasv}}$	3.57	5.23	2.03	1.88	2.85	0.82
$\mathcal{L}_{\text{cont}}^{\text{sasv}} + \mathcal{L}_{\text{id}_1}^{\text{sasv}}$	<b>1.25</b>	<b>1.27</b>	<b>1.23</b>	<b>1.49</b>	<b>1.79</b>	<b>1.28</b>
$\mathcal{L}_{\text{cont}}^{\text{sasv}} + \mathcal{L}_{\text{id}_2}^{\text{sasv}}$	3.00	3.25	2.81	2.39	3.00	1.81

CS data), a scheduler cycle size of 20 epochs and a maximum learning rate of 1e-5 which was decreased by 0.5 every three cycles. In the case of ASVspoof 2019 LA bona fide and CS data, we used a batch size of 32 (2 utterances per 8 speakers for bona fide and CS data). For Stage 3, we used the same settings as for ASVspoof 2019 LA data in Stage 2, and data for 16 randomly chosen spoofing attacks instead of 16 CS utterances.

## 4. Results

### 4.1. Multi-stage training and CS results

Results are shown in Table 1. The comparison of results in rows 1 and 3 shows the benefit of Stage 1 ASV pre-training when used with Stage 3 fine-tuning. We report four types of results, using two model architectures and two Stage 3 training data configurations. The objective is to verify whether the additional use of development partition of ASVspoof2019LA in Stage 3 can be advantageous. SASV-EERs improved in all four columns, where it was more advantageous when stage 3 only included the train partition (*trn*). Rows 4-7 show results for Stage 2 CS-training, with or without Stage 1 and Stage 3 training. Results in row 4 show that Stage 2 is more beneficial than Stage 1; performance improvements are observed for both models. This makes sense since data used for Stage 1 can be viewed as a *subset* of the data used for Stage 2. The comparison of results in rows 5 and 7 show that Stage 1 training is complementary to Stage 2.

Rows 8 to 11 show results using Stage 2 training with ASVspoof 2019 LA CS instead of VoxCeleb2 CS. The comparison of results in rows 8 and 9 with those in rows 4 and 5 shows similar results with no fine-tuning, but worse results with; fine tuning using VoxCeleb, which contains data collected from a far greater number of speakers, is the most beneficial. The comparison of results in rows 10 and 11 to rows 8 and 9 show slightly improved performance when ASVspoof 2019 LA data is used in Stage 2 and combined with VoxCeleb data in Stage 1. Even so, results are slightly worse than corresponding results in rows 4-7 for which VoxCeleb data is used for Stage 2. These results show that the use of data collected from a larger number of speakers leads to better speaker discrimination and SASV performance.

Results for the two models show similar trends. The use of all three stages together with VoxCeleb data for Stage 2 (row 7), results in the lowest SASV-EER for both models; the only exception is for the SKA-TDNN model fine-tuned in Stage 3 with both train and development sets. The best result of 1.06% is produced with the MFA-Conformer model when trained using all three stages and when fine-tuned using both train and development sets of the ASVspoof 2019 LA database. In summary, Stage 1 pre-training with VoxCeleb data, followed by Stage 2 training with CS data, and then fine-tuning with in-domain ASVspoof 2019 LA data generally leads to the best

Table 3: Comparison with previous works in the literature. With Stage 1 and Stages 1-2 denote the results with pre-training (Stage 1) and sequential CS-training (Stages 1-2), respectively.

System	Type	SASV	SV	SPF
ECAPA-TDNN [42]	ASV	23.83	1.63	30.75
AASIST [43]	CM	24.38	49.24	0.67
SASV Baseline [44]	Score-level Fusion	1.71	1.66	1.76
[15]	Emb. level fusion	0.28	0.28	0.28
[18]	Joint-optimization	1.53	2.44	0.75
[17]	Single system	4.86	8.06	0.50
<b>Ours - Stage 3 only</b>	Single system	11.47	13.46	10.30
<i>with Stage 1</i>	Single system	2.13	3.43	1.08
<i>with Stages 1-2</i>	Single system	<b>1.19</b>	<b>1.83</b>	<b>0.58</b>

performance.

### 4.2. Loss function comparison

Table 2 shows the results adopting Stages 1 and 3, with different loss functions and their combination. When using only one of the three loss functions, the trends differ for each model. The use of contrastive loss ( $\mathcal{L}_{\text{cont}}^{\text{sasv}}$ ) with integrated identification loss ( $\mathcal{L}_{\text{id}_1}^{\text{sasv}}$ ), leads to the best performance for both models. The comparison of identification losses ( $\mathcal{L}_{\text{id}_1}^{\text{sasv}}$  and  $\mathcal{L}_{\text{id}_2}^{\text{sasv}}$ ) shows that integrated outperformed its counterpart in most cases.

### 4.3. Performance comparison

Table 3 shows a comparison of results for the proposed single SASV system to those for competing systems reported in the literature. The first two rows show results for the individual ASV and CM subsystems used by SASV challenge baselines [8]. Rows 3 to 6 show results for the SASV B1 baseline<sup>4</sup>, the best embedding-level fusion system, the best jointly-optimized system, and best single end-to-end system reported in the literature.<sup>5</sup> Results for our best system (bottom row) correspond to an 84% relative improvement in SASV-EER compared to the best single system for which the SASV-EER is 4.86%. This pushes the state-of-the-art in single SASV approaches dramatically. It also outperforms the best jointly-optimized solution.

## 5. Conclusions

We present a new integrated SASV embedding extractor that achieves an SASV-EER of 1.06%, significantly surpassing previous best results for such systems. This approach bridges the performance gap between single integrated SASV systems and fusion-based methods that combine pre-trained ASV and CM models. Our multi-stage training solution benefits from pre-training with large amount of data also with CS, which addresses the scarcity of spoofed data and the imbalance with genuine training data.

## 6. Acknowledgements

This work is partially supported by the COMPA grant funded by the Korea government (MSIT and Police) (No. RS-2023-00235082), Academy of Finland (Decision No. 349605, project ‘‘SPEECHFAKES’’), JST CREST (JPMJCR18A6, JPMJCR20D3), and MEXT KAKENHI (21K17775, 21H04906).

<sup>4</sup>The 2022 SASV challenge B1 baseline with softmax applied to the CM output which improves results notably.

<sup>5</sup>Better results are reported in the literature, but these correspond to *ensemble* systems. The comparisons presented here are restricted to systems which employ at most one CM and one ASV subsystem.

## 7. References

- [1] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” *Speech communication*, vol. 52, no. 1, 2010.
- [2] Z. Bai and X.-L. Zhang, “Speaker recognition based on deep learning: An overview,” *Neural Networks*, vol. 140, 2021.
- [3] Y. Zhang, Z. Lv, H. Wu et al., “Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification,” in *Proc. INTERSPEECH*, 2022.
- [4] Z. Chen, S. Chen, Y. Wu et al., “Large-scale self-supervised speech representation learning for automatic speaker verification,” in *Proc. ICASSP*, 2022.
- [5] T. Kinnunen, M. Sahidullah, H. Delgado et al., “The ASVspoof 2017 Challenge: Assessing the limits of replay spoofing attack detection,” in *Proc. INTERSPEECH*, 2017.
- [6] X. Wang, J. Yamagishi, M. Todisco et al., “ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” *Computer Speech & Language*, vol. 64, 2020.
- [7] J. Yamagishi, X. Wang, M. Todisco et al., “ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection,” in *Proc. the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, 2021.
- [8] J.-w. Jung, H. Tak, H.-j. Shim et al., “SASV 2022: The first spoofing-aware speaker verification challenge,” in *Proc. INTERSPEECH*, 2022.
- [9] H.-j. Shim, J.-w. Jung, J.-h. Kim and H.-j. Yu, “Integrated replay spoofing-aware text-independent speaker verification,” *Applied Sciences*, vol. 10, no. 18, 2020.
- [10] 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, 2020.
- [11] M. Todisco, H. Delgado, K. A. Lee et al., “Integrated presentation attack detection and automatic speaker verification: Common features and gaussian back-end fusion,” in *Proc. INTERSPEECH*, 2018.
- [12] Y. Zhang, G. Zhu and Z. Duan, “A probabilistic fusion framework for spoofing aware speaker verification,” in *Proc. Speaker Odyssey*, 2022.
- [13] J. Lin, T. Chen, J. Huang et al., “The CLIPS system for 2022 spoofing-aware speaker verification challenge,” in *Proc. INTERSPEECH*, 2022.
- [14] X. Wang, X. Qin, Y. Wang et al., “The DKU-OPPO system for the 2022 spoofing-aware speaker verification challenge,” in *Proc. INTERSPEECH*, 2022.
- [15] J.-H. Choi, J.-Y. Yang, Y. R. Jeoung and C. J.-H., “HYU submission for the SASV challenge 2022: Reforming speaker embeddings with spoofing-aware conditioning,” in *Proc. INTERSPEECH*, 2022.
- [16] W. H. Kang, J. Alam and A. Fathan, “End-to-end framework for spoof-aware speaker verification,” in *Proc. INTERSPEECH*, 2022.
- [17] Z. Teng, Q. Fu, J. White et al., “SA-SASV: An end-to-end spoof-aggregated spoofing-aware speaker verification system,” in *Proc. INTERSPEECH*, 2022.
- [18] W. Ge, H. Tak, M. Todisco and N. Evans, “On the potential of jointly-optimised solutions to spoofing attack detection and automatic speaker verification,” in *Proc. IberSPEECH*, 2022.
- [19] A. Alenin, N. Torgashov, A. Okhotnikov et al., “A subnetwork approach for spoofing aware speaker verification,” in *Proc. INTERSPEECH*, 2022.
- [20] P. Zhang, P. Hu and X. Zhang, “Norm-constrained score-level ensemble for spoofing aware speaker verification,” in *Proc. INTERSPEECH*, 2022.
- [21] L. Zhang, Y. Li, H. Zhao and L. Xie, “Backend ensemble for speaker verification and spoofing countermeasure,” in *Proc. INTERSPEECH*, 2022.
- [22] J. S. Chung, A. Nagrani and A. Zisserman, “Voxceleb2: Deep speaker recognition,” in *Proc. INTERSPEECH*, 2018.
- [23] X. Wang and J. Yamagishi, “Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders,” in *Proc. ICASSP*, 2023.
- [24] S. H. Mun, J.-w. Jung, M. H. Han and N. S. Kim, “Frequency and multi-scale selective kernel attention for speaker verification,” in *Proc. SLT*, 2023.
- [25] W. J. Holmes, “Copy synthesis of female speech using the JSRU parallel formant synthesiser,” in *Proc. EUROSPEECH*, 1989, pp. 2513–2516.
- [26] 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*, 2019.
- [27] H. S. Heo, B.-J. Lee, J. Huh and J. S. Chung, “Clova baseline system for the voxceleb speaker recognition challenge 2020,” *arXiv preprint arXiv:2009.14153*, 2020.
- [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*, 2021.
- [29] M. Pal, D. Paul and G. Saha, “Synthetic speech detection using fundamental frequency variation and spectral features,” *Computer Speech & Language*, vol. 48, 2018.
- [30] J. Frank and L. Schönherr, “Wavefake: A data set to facilitate audio deepfake detection,” in *Proc. Neural Information Processing Systems Datasets and Benchmarks*, 2021.
- [31] Z. Wu, X. Xiao, E. S. Chng and H. Li, “Synthetic speech detection using temporal modulation feature,” in *Proc. ICASSP*, 2013.
- [32] A. Sizov, E. Khoury, T. Kinnunen et al., “Joint speaker verification and antispoofing in the i-vector space,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, 2015.
- [33] I. Saratxaga, J. Sanchez, Z. Wu et al., “Synthetic speech detection using phase information,” *Speech Communication*, 2016.
- [34] C. Sun, S. Jia, S. Hou et al., “Exposing ai-synthesized human voices using neural vocoder artifacts,” *arXiv preprint arXiv:2302.09198*, 2023.
- [35] J. Kong, J. Kim and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proc. NIPS*, 2020, vol. 33.
- [36] R. Prenger, R. Valle and B. Catanzaro, “WaveGlow: A flow-based generative network for speech synthesis,” in *Proc. ICASSP*, 2019.
- [37] X. Wang, S. Takaki and J. Yamagishi, “Neural source-filter-based waveform model for statistical parametric speech synthesis,” in *Proc. ICASSP*, 2019.
- [38] J. Engel, L. Hantrakul, C. Gu and A. Roberts, “DDSP: Differentiable digital signal processing,” in *Proc. ICLR*, 2020.
- [39] J. S. Chung, J. Huh, S. Mun et al., “In defence of metric learning for speaker recognition,” in *Proc. INTERSPEECH*, 2020.
- [40] R. Caruana, *Multitask learning*, Springer, 1998.
- [41] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proc. ICLR*, 2017.
- [42] B. Desplanques, J. Thienpondt and K. Demuynck, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” in *Proc. INTERSPEECH*, 2020.
- [43] J.-w. Jung, H.-S. Heo, H. Tak et al., “Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks,” in *Proc. ICASSP*, 2022.
- [44] H.-j. Shim, H. Tak, X. Liu et al., “Baseline systems for the first spoofing-aware speaker verification challenge: score and embedding fusion,” in *Proc. Speaker Odyssey*, 2022.