



## SZU-AFS Antispoofing System for the ASVspoof 5 Challenge

Yuxiong Xu<sup>1</sup>, Jiafeng Zhong<sup>1</sup>, Sengui Zheng<sup>2</sup>, Zefeng Liu<sup>1</sup>, Bin Li<sup>1\*</sup>

<sup>1</sup>Guangdong Provincial Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Media Security, and SZU-AFS Joint Innovation Center for AI Technology, Shenzhen University, Shenzhen, China

<sup>2</sup>Afirstsoft Technology Group co., Ltd., Shenzhen, China

{xuyuxiong2022, jiafengzhong2022}@email.szu.edu.cn, 1225620446@qq.com

liuzefeng2021@email.szu.edu.cn, libin@szu.edu.cn

### Abstract

This paper presents the SZU-AFS anti-spoofing system, designed for Track 1 of the ASVspoof 5 Challenge under open conditions. The system is built with four stages: selecting a baseline model, exploring effective data augmentation (DA) methods for fine-tuning, applying a co-enhancement strategy based on gradient norm aware minimization (GAM) for secondary fine-tuning, and fusing logits scores from the two best-performing fine-tuned models. The system utilizes the Wav2Vec2 front-end feature extractor and the AASIST back-end classifier as the baseline model. During model fine-tuning, three distinct DA policies have been investigated: single-DA, random-DA, and cascade-DA. Moreover, the employed GAM-based co-enhancement strategy, designed to fine-tune the augmented model at both data and optimizer levels, helps the Adam optimizer find flatter minima, thereby boosting model generalization. Overall, the final fusion system achieves a minDCF of 0.115 and an EER of 4.04% on the *evaluation* set.

### 1. Introduction

Recent advancements in Artificial Intelligence Generated Content (AIGC) have significantly enhanced the naturalness, fidelity, and variety of speech. Unfortunately, this progress has resulted in a proliferation of forgeries that can be almost indistinguishable from authentic speech to the human auditory system. Concurrently, automatic speaker verification (ASV) systems have become increasingly susceptible to spoofing and deepfake attacks, in which attackers produce convincingly realistic simulations of the target speaker's voice [1]. The potential misuse of spoofed speech presents significant societal risks. Therefore, developing a robust and generalizable anti-spoofing system to counter these threats has emerged as a critical research imperative.

The ASVspoof challenges [2, 3, 4, 5, 6] have significantly boosted interest in developing robust detection solutions for spoofing and deepfake attacks, thereby enhancing the security and reliability of ASV systems. These challenges provide standardized benchmark protocols and comprehensive evaluation datasets. What's more, the last four ASVspoof challenges [2, 3, 4, 5] have prompted the proposal of numerous innovative spoofing detection methods [7, 8, 9, 10].

Held in 2024, the ASVspoof 5 Challenge [6] presents two distinct conditions, open and closed, for both Track 1 which focuses on standalone speech deepfake detection, and Track 2

\*Corresponding author

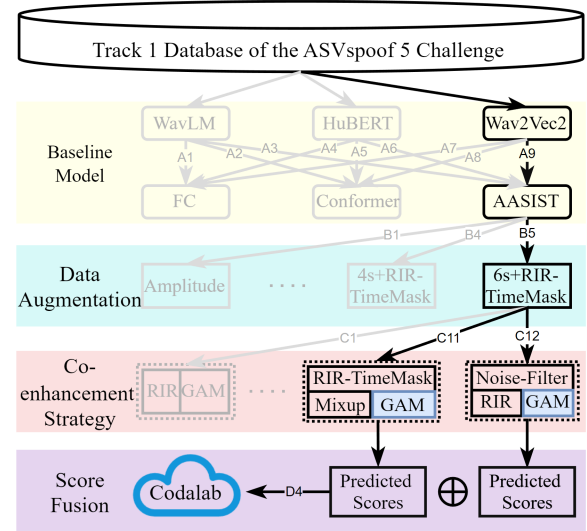


Figure 1: Illustration of the SZU-AFS anti-spoofing system. The colored boxes represent four stages of the system, with each stage labeled by model IDs from A to D. The best-performing model in each stage and its ID number are presented in bold. First, a baseline model (A9) was selected, combining the Wav2Vec2 feature extractor with the AASIST classifier. The A9 model was then fine-tuned using the RIR-TimeMask method to obtain the best-augmented model (B5), which was subsequently further fine-tuned using a GAM-based co-enhancement strategy. Finally, the logits scores from the C11 and C12 models were fused using an average score-level fusion method, and the results were submitted for evaluation on the CodaLab platform.

which is dedicated to spoofing-robust automatic speaker verification. Under the closed conditions, participants are restricted from using specified data protocols. Conversely, the open conditions offer greater flexibility, allowing participants to utilize external data and pre-trained self-supervised models, provided there is no overlap between training data (i.e., that used for training foundational models) and evaluation data. Track 1 is similar to the DF track of the ASVspoof 2021 Challenge, reflecting a scenario in which an attacker has access to a targeted victim's voice data, such as data posted on social media. In this scenario, the *evaluation* set contained data processed with conventional codecs or modern neural codecs. Track 2, similar to the LA sub-task from previous ASVspoof challenges, is pred-

icated on a telephony scenario where synthetic and converted speech is directly injected into a communication system without any acoustic propagation.

The ASVspoof 5 Challenge introduces significant changes in source data, attack types, and evaluation metrics. The source data, extracted from the Multilingual LibriSpeech English partition [11], includes a vastly greater number of speakers than previous ASVspoof databases. Notably, the spoofing attacks in the *training*, *development*, and *evaluation* sets are entirely disjoint. As shown in Table 1, the *training* dataset is used to adjust model parameters, while the *development* dataset is used to tune and evaluate performance. The *progress* set initially assesses the detection model’s performance, allowing participants up to four submissions per day via the CodaLab platform. The *evaluation* set tests its generalizability, with only one submission allowed per team. New evaluation metrics, minDCF [12] for Track 1 and agnostic DCF [13] for Track 2, have been introduced to better assess anti-spoofing systems.

This paper presents the SZU-AFS anti-spoofing system for Track 1 under open conditions. Its design diagram is illustrated in Figure 1, where model IDs are labeled from A to D, with numbers indicating their respective versions. This system has four stages: baseline model selection, exploration of effective data augmentation (DA) methods for fine-tuning, application of a co-enhancement strategy utilizing gradient norm aware minimization (GAM) for secondary fine-tuning, and fusion of logits scores from the two top-performing models. Specifically, comparative experimental analysis was conducted first by combining three pre-trained models with three distinct classifiers to select an appropriate baseline model. We have selected the pre-trained Wav2Vec2 model [14] as the feature extractor, coupled with the AASIST classifier [15], to serve as the baseline model (A9). Secondly, we have proposed three DA policies to explore the effectiveness of various DA methods: single-DA, random-DA, and cascade-DA. The best-performing model is the one fine-tuned by augmented data generated by sequentially applying room impulse response (RIR) noise and time masking (TimeMask) method, resulting in an augmented model (B5). Next, we employed a GAM-based co-enhancement strategy to consider data and optimizer simultaneously to enhance model generalizability. With this strategy, the B5 model has been fine-tuned by combining various DA methods with the GAM method, resulting in the C11 and C12 models as the two best-performing fine-tuned models. Finally, we have fused the predicted logits scores from the C11 and C12 models using an average score-level fusion method to generate final evaluation scores, constituting system D4.

This paper is organized as follows: Section 2 elaborates the core modules of the SZU-AFS system, including the baseline model, the three DA policies, the GAM-based co-enhancement strategy, and the score-level fusion. Implementation details regarding the dataset information and model hyperparameters are provided in Section 3. Section 4 provides experimental results and analysis. Conclusions are drawn in Section 5.

## 2. Methodology

The SZU-AFS anti-spoofing system consists of four stages detailed in separate subsections: baseline model, data augmentation, gradient norm aware minimization (GAM)-based co-enhancement strategy, and score-level fusion. Note that we trained ten different baseline models, six augmented models, and eight models using various DA and GAM methods. Model IDs are labeled A to D, and numbers indicate versions.

### 2.1. Baseline Model

#### 2.1.1. Front-end feature extractor

Given the urgent need to improve the generalizability of spoofing detection systems, speech self-supervised models have gained increasing attention. Prior research shows that using speech self-supervised models as the front-end feature extractors and the back-end classifier, can substantially improve the generalization of spoofing detection models [16, 17, 18, 19].

We have used the self-supervised WavLM-Base<sup>1</sup> [20], HuBERT-Base<sup>2</sup> [21], and Wav2Vec2-Large<sup>3</sup> [14] as front-end feature extractors instead of conventional handcrafted acoustic features, such as linear frequency cepstral coefficients and mel-spectrograms. The self-supervised learning models extract speech representations or embeddings from the raw waveform.

#### 2.1.2. Back-end classifier

The back-end classifiers of the latest spoofing detection systems mainly adopt deep learning methods [22, 23, 24], significantly outperforming traditional classifiers such as support vector machine and Gaussian mixture model [25, 26]. We have tried three representative classifiers combined with front-end pre-trained models, detailed as follows:

- **Fully connected (FC) classifier** [22]: This classifier combines a global average pooling layer, followed by a neural network with three fully connected layers employing LeakyReLU activation functions. It ends with a linear output layer for binary classification.
- **Conformer classifier** [23]: This classifier combines a convolutional neural network and a Transformer network for spoofing detection. It comprises four blocks, each with four attention heads and a kernel size of 31, totaling 2.4 million parameters.
- **AASIST classifier** [24]: This classifier combines a RawNet2-based encoder [27] and a graph network module. Specifically, it removes the Sinc convolutional layer-based front-end from the RawNet2-based encoder.

#### 2.1.3. Model selection

As shown in Table 2, we have evaluated the detection performance of the A1-A10 models using the development set of ASVspoof 5. The A1-A9 models are combinations of three pre-trained models with three different classifiers, while the A10 model combines the Wav2Vec2 pre-trained model with all classifiers, generating predictive scores by processing concatenated features through a linear layer. According to experimental results, we have selected the A9 model as the baseline by utilizing a Wav2Vec2-based front-end feature extractor paired with an AASIST-based back-end classifier.

### 2.2. Primary Fine-tuning with Data Augmentation

To enhance the generalization performance, we have conducted experiments with three DA policies: single-DA, random-DA, and cascade-DA, to fine-tune the baseline model. The three DA policies, as depicted in Figure 2, are detailed as follows.

<sup>1</sup><https://github.com/microsoft/unilm/blob/master/wavlm>

<sup>2</sup><https://github.com/facebookresearch/fairseq/tree/main/examples/hubert>

<sup>3</sup><https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec>

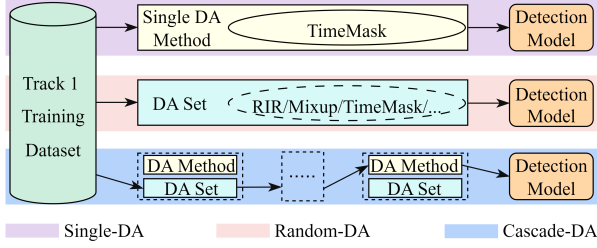


Figure 2: Illustration of the three different DA policies. To enhance the generalization abilities of the A9 model, we experiment with three distinct DA policies, including single-DA, random-DA, and cascade-DA.

### 2.2.1. Single-DA policy

A specific DA method is applied to all original training data for the single-DA policy. The details of the DA methods employed are described below:

- **RIR**<sup>4</sup>: The room impulse response (RIR) captures the acoustic characteristics of a room or an environment. A noise clip is randomly selected from the RIR database and superimposed onto the original training speech, with the intensity randomly varying between 20% and 80%.
- **RawBoost** [28]: RawBoost incorporates 3 distinct types of noise: linear and non-linear convolutive (LnL) noise, stationary signal-independent additive (SSI) noise, and impulsive signal-dependent additive (ISD) noise.
- **Signal companding**: The a-law and  $\mu$ -law are signal companding methods developed to enable the transmission of signals with a large dynamic range through systems with limited dynamic range capabilities. During the enhancement of the input speech, either a-law or  $\mu$ -law is randomly selected.
- **TimeMask**: For the input speech, consecutive time steps from  $t_0$  to  $t_0 + t$  are set to zero. The duration  $t$  is uniformly selected from 0 to  $T$ , and the starting point  $t_0$  is randomly chosen from the interval  $[0, \tau - t)$ . Here,  $\tau$  represents the total number of time steps, and  $T$  varies randomly between 20% and 50% of  $\tau$ .
- **Mixup** [29]: Mixup regularization involves training the model using a set of mixed speech utterances and labels, rather than the original training data, with the interpolation parameter sampled from a symmetric  $Beta(\sigma, \sigma)$  distribution, where  $\sigma = 1.0$ .
- **Amplitude**: Amplitude enhancement involves selecting two speech utterances from the same speaker and label, mixing their amplitude spectra with a certain probability, and then applying inverse Fourier transformation with the corresponding phase spectra to obtain the enhanced utterances.

### 2.2.2. Random-DA policy

Unlike a single-DA strategy, the random-DA policy involves randomly selecting an augmentation method from a DA set for each utterance of the original training data. More specifically, we used three DA sets:

- **Noise set**: This set contains 3 noise-based DA methods from the audiomentations<sup>5</sup> library, with corresponding

modules named AddColorNoise, AddGaussianNoise, and AddGaussianSNR.

- **Filter set**: This set contains 7 filter-based DA methods from the audiomentations library, with corresponding modules named BandPassFilter, BandStopFilter, HighPassFilter, HighShelfFilter, LowPassFilter, LowShelfFilter, and PeakingFilter.
- **Mix set**: This set contains 13 DA methods of mixed types from the audiomentations library, with corresponding modules named AddGaussianNoise, AirAbsorption, Aliasing, BandPassFilter, Shift, PitchShift, HighPassFilter, LowPassFilter, PolarityInversion, PeakingFilter, TimeStretch, TimeMask, and TanhDistortion.

### 2.2.3. Cascade-DA policy

The cascade-DA policy encourages selecting two or more DA methods in a sequential cascade manner to enhance the original training data progressively. Three types of cascade-DA methods are given below:

- **RIR-TimeMask**: RIR-TimeMask consists of a two-level cascade of DA methods, sequentially adding RIR noise and TimeMask method to the original training data.
- **LnL-ISD**: LnL-ISD consists of a two-level cascade of DA methods, sequentially adding LnL and ISD noise to the original training data. Both LnL noise and ISD noise are derived from the RawBoost method.
- **Noise-Filter**: Noise-Filter consists of a two-level cascade of DA sets, sequentially applying one method randomly selected from the noise set and another from the filter set to enhance the original training data.

Note that the RIRTimeMask method is used in the primary fine-tuning stage, while LnL-ISD, Noise-Filter, and combinations of cascade-DA methods are used in the secondary fine-tuning stage.

### 2.2.4. Model selection

The A9 model is fine-tuned using only the on-the-fly augmented data. We have evaluated the detection performance of six models (B1-B6, which are shown in Table 3) with different DA methods, using the *progress* set of ASvspoof 5. Specifically, we have fine-tuned the A9 model using distinct DA methods, including Amplitude, a-law or  $\mu$ -law, Mix, and RIR-TimeMask. The B4-B6 models share the same augmentation methods but vary in the number of input speech samples used for training: 64,600, 96,000, and 128,000, respectively. Following experimental analysis, the B5 model has been chosen for further investigation.

## 2.3. Secondary Fine-tuning with GAM-based Co-enhancement Strategy

Unlike DA methods, which focus on increasing the diversity of training data, the GAM method is an optimization approach for enhancing model generalization. To alleviate this issue, the fine-tuning process has been divided into two stages: a primary stage without GAM, as described in the previous subsection, and a secondary stage with DA and GAM co-enhancement, as illustrated in this subsection.

### 2.3.1. Gradient norm aware minimization

Sharpness-aware minimization (SAM) [30] and its variants [31] are representative training algorithms to seek flat minima for

<sup>4</sup><https://www.openslr.org/28/>

<sup>5</sup><https://github.com/iver56/audiomentations>

better generalization. Shim et al. [32] employed SAM and its variants in spoofing detection, improving model generalization. Inspired by this, we exploit a recently proposed optimization method, gradient norm aware minimization (GAM) [33].

GAM seeks flat minima with uniformly small curvature across all directions in the parameter space. Specifically, it improves the generalization of models trained with the Adam optimizer by optimizing first-order flatness, which controls the maximum gradient norm in the neighborhood of minima.

Let  $\theta \in \Theta \subseteq \mathbb{R}^d$  denote the parameters of the B5 model. The Adam optimizer is then described as follows:

$$\theta_{t+1} = \theta_t - \eta g_t, \quad (1)$$

where  $t$  is the time step,  $\eta$  is the learning rate, and  $g_t$  is the loss gradient. For the first-order flatness  $R_\rho^1(\theta)$ , it could be computed by:

$$R_\rho^1(\theta) \triangleq \rho \cdot \max_{\theta' \in B(\theta, \rho)} \|\nabla \hat{L}(\theta')\|, \quad (2)$$

where  $\hat{L}(\theta) = \sum_{i=1}^n \ell(\theta, x_i, y_i)$  denotes the empirical loss function,  $x_i$  and  $y_i$  denote the  $i$ -th speech sample and its corresponding label, respectively.  $\rho > 0$  is the perturbation radius that controls the magnitude of the neighborhood, and  $B(\theta, \rho)$  denotes the open ball of radius  $\rho$  centered at the parameter  $\theta$  in the Euclidean space. For detailed derivation, see Appendix A of [33]. The key to optimizing generalization error with GAM is controlling the loss function  $\hat{L}(\theta)$  and first-order flatness  $R_\rho^1(\theta)$ . The pseudocode of the whole optimization procedure is shown in Algorithm 1.

---

**Algorithm 1** Gradient norm Aware Minimization (GAM)

---

**Input:** Batch size  $b$ , learning rate  $\eta_t$ , perturbation radius  $\rho_t$ , trade-off coefficient  $\alpha$ , small constant  $\xi$

- 1:  $t \leftarrow 0, \theta_0 \leftarrow$  initial parameters
- 2: **while**  $\theta_t$  not converged **do**
- 3:   Sample  $W_t$  from the training data with  $b$  instances
- 4:   Calculate the empirical loss gradient  $\nabla \hat{L}(\theta_t)$ :  
 $h_t^{\text{loss}} \leftarrow \nabla \hat{L}(\theta_t)$
- 5:   Calculate the perturbed gradient using the loss gradient  $\nabla \hat{L}_{W_t}(\theta_t)$  of the sample  $W_t$  and the Hessian matrix  $\nabla^2 \hat{L}_{W_t}(\theta_t)$ :  
 $f_t \leftarrow \nabla^2 \hat{L}_{W_t}(\theta_t) \cdot \frac{\nabla \hat{L}_{W_t}(\theta_t)}{\|\nabla \hat{L}_{W_t}(\theta_t)\| + \xi}$
- 6:   Calculate the adversarial parameters adjusted via the perturbed gradient:  
 $\theta_t^{\text{adv}} \leftarrow \theta_t + \rho_t \cdot \frac{f_t}{\|f_t\| + \xi}$
- 7:   Calculate the norm gradient  $\nabla R_{\rho_t}^{(1)}(\theta_t)$ :  
 $h_t^{\text{norm}} \leftarrow \rho_t \cdot \nabla^2 \hat{L}_{W_t}(\theta_t^{\text{adv}}) \cdot \frac{\nabla \hat{L}_{W_t}(\theta_t^{\text{adv}})}{\|\nabla \hat{L}_{W_t}(\theta_t^{\text{adv}})\| + \xi}$
- 8:    $\theta_{t+1} \leftarrow \theta_t - \eta_t(h_t^{\text{loss}} + \alpha h_t^{\text{norm}})$
- 9:    $t \leftarrow t + 1$
- 10: **end while**
- 11: **return**  $\theta_t$ .

---

### 2.3.2. Co-enhancement strategy

The GAM-based co-enhancement strategy involves data augmentation of the input speech and combines the GAM method with the Adam optimizer to further fine-tune the DA-augmented baseline model (B5). Unlike the primary fine-tuning with DA methods, this strategy has explored more efficient two-level and three-level DA methods, combined with RIR or TimeMask, to process the original training data. Specifically, we have combined eight different DA methods with GAM: C1 (RIR), C2

Table 1: Summary of ASVspoof 5 Track 1 database. “Spr.” denotes the number of speakers, while “Train.”, “Dev.”, “Prog.”, “Eval.” refer to the training, development, progress, and evaluation sets, respectively.

Sets	Spr.	Attack Types	Utterances		
			Bona fide	Spoofed	Total
Train.	400	A1-A8	18,797	163,560	182,357
Dev.	785	A9-A16	31,334	109,616	140,950
Prog.	—	—	—	—	40,765
Eval.	737	A17-A32	395,924	138,688	680,774

(a-law or  $\mu$ -law), C3 and C4 (Mix), C5 and C6 (LnL-ISD), C7 (RIR + Mix), C8 and C9 (RIR-TimeMask), C10 and C11 (RIR-TimeMask + Mixup), and C12 (RIR + Noise-Filter). As shown in Table 4, we have evaluated the detection performance of models C1-C12 using the *progress* set of ASVspoof 5. Experimental analysis indicates that the C11 and C12 models are the two best-performing models in terms of minDCF.

### 2.4. Score-level Fusion

The individual model scores have been directly output as logits from the linear layer without applying min-max normalization. Building on this, we have utilized an average score-level fusion method, where the predicted scores from each model have been summed and averaged to determine the final prediction score.

As shown in Table 5, we have evaluated the detection performance of fused models D1-D4 on either the *progress* or *evaluation* sets of ASVspoof 5. Specifically, we have tested four fused models: D1 (B4 + B5), D2 (B1 to B6), D3 (C8 + C9), and D4 (C11 + C12). Among these models, we have selected the best-performing fused system, D4, for submission to the evaluation phase.

## 3. Experimental Setup

### 3.1. Datasets and Metrics

#### 3.1.1. Datasets

This paper focuses on the Track 1 stand-alone speech deepfake detection task of ASVspoof 5, with a summary of the Track 1 database provided in Table 1. The dataset contains 1,044,846 utterances, each encoded as a 16 kHz, 16-bit FLAC file. The *training* and *development* sets each contain spoofed speech generated by 8 different text-to-speech (TTS) or voice conversion (VC) methods. In contrast, the *evaluation* set includes spoofed speech from 16 diverse attack methods, including TTS, VC, and, for the first time, adversarial attacks. The *evaluation* set contains more than twice the number of samples as the combined *training* and *development* sets, making detection significantly more challenging. Notably, the *progress* set is a subset of the *evaluation* set.

#### 3.1.2. Metrics

Different from previous ASVspoof challenges, ASVspoof 5 Challenge uses the minDCF as the primary metric for the comparison of spoofing countermeasures, with the cost of log-likelihood ratio ( $C_{llr}$ ) [34] and the equal error rate (EER) as a secondary metrics. Accuracy (ACC) was introduced to evaluate the detection model’s performance on the development set. In contrast, EER provides a more suitable measure of perfor-

Table 2: *Performance in Accuracy (%) and EER (%) of different baseline models on the Track 1 development set. The highlighted model was selected for further fine-tuning to enhance its generalizability.*

Model ID	Feature Extractor	Back-end Classifier	Accuracy (%)	EER (%)	
A1	WavLM	FC	54.56	40.00	
A2		Conformer	67.80	43.50	
A3		AASIST	77.25	42.70	
A4	HuBERT	FC	73.12	19.43	
A5		Conformer	78.31	9.56	
A6		AASIST	81.58	7.81	
A7	Wav2Vec2	FC	<b>91.49</b>	2.17	
A8		Conformer	81.81	6.50	
A9		AASIST	87.64	<b>1.55</b>	
A10		FC + AASIST + Conformer	88.56	2.04	

mance when the data is limited or imbalanced. Thus, EER is better suited than ACC for evaluating spoof detection models.

The normalised detection cost function (DCF) is:

$$\text{DCF}(\tau_{cm}) = \beta \cdot P_{\text{miss}}^{\text{cm}}(\tau_{cm}) + P_{\text{fa}}^{\text{cm}}(\tau_{cm}), \beta = \frac{C_{\text{miss}}}{C_{\text{fa}}} \cdot \frac{1 - \pi_{\text{spf}}}{\pi_{\text{spf}}}, \quad (3)$$

where  $\tau_{cm}$  is the detection threshold,  $\pi_{\text{spf}}$  is asserted prior probability of spoofing attack, and  $C_{\text{miss}}$  and  $C_{\text{fa}}$  are the costs of a miss and a false alarm, respectively. The following parameters were used for the ASVspoof 5 challenge evaluation:  $C_{\text{miss}} = 1$ ,  $C_{\text{fa}} = 10$ ,  $\pi_{\text{spf}} = 0.05$ , and  $\beta \approx 1.90$ . The normalized DCF in (3) is used to compute the minimum DCF, defined as  $\text{minDCF} = \min_{\tau_{cm}} \text{DCF}(\tau_{cm})$ .

### 3.2. Training Details

In our experiments, the following parameters were kept consistent. We used the Adam optimizer ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ) [35], with an initial learning rate of  $5 \times 10^{-6}$ , controlled by a cosine annealing scheduler with a minimum learning rate of  $1 \times 10^{-8}$ , and a maximum of 100 training epochs. The training was conducted using conventional cross-entropy loss, with early stopping applied if the development set loss did not improve within ten epochs. All experiments were executed on two NVIDIA A100 GPUs. The training epochs for the models used to obtain the D4 system were as follows: 12 epochs (A9), 4 epochs (B5), 2 epochs (C11), and 5 epochs (C12). The training time required for the combined DA and GAM method is approximately three times that of the regular DA method alone. The results table highlights the best-performing values in bold for each column.

## 4. Results and Analysis

### 4.1. Comparison Analysis of Different Baseline Models

We integrated three pre-trained models with three classifiers to assess the necessity of different baseline models, testing their performance on the Track 1 *development* set. The training and development data were truncated or padded to 64,600 sample points to accommodate GPU memory constraints.

The accuracy and EER for different baseline models are reported in Table 2. As indicated in Table 2, we observe:

- Among the different feature extractors, Wav2Vec2-based

detection models (e.g., A7, A8, A9, and A10) achieved higher accuracy and lower EER, which indicates their better effectiveness than WavLM-based and HuBERT-based detection models. This result is due to the fact that both the WavLM and HuBERT pre-trained models use the base version, which contains fewer than one-third of the learnable parameters in the Wav2Vec2-Large model.

- Among the different classifiers, AASIST proved to be more competitive than others (e.g., FC and Conformer) when paired with various feature extractors, further confirming its excellent performance in speech spoofing detection. Furthermore, leveraging the Wav2Vec2 feature extractor, we concatenated the final predictive outputs from three classifiers for classification. However, the A10 model’s results fell between the individual classifiers’ results, providing no improvement. Thus, using only the best classifier is sufficient.
- While the A9 model’s accuracy is slightly lower at 87% compared to the A7 and A10 models, it achieves the lowest EER at 1.55%. Owing to its outstanding EER performance, the A9 model has been selected for further experimental exploration.

### 4.2. Comparison Analysis of Different DA

Table 3 shows the effectiveness of various DA methods during the progress phase of Track 1. Compared with the B1 and B3 models, the B2 and B4 models achieved lower minDCF and EER. The A9 model presents superior detection performance when fine-tuned with signal compression (a-law or  $\mu$ -law), RIR noise, and TimeMask. However, the B4 model exhibited a higher  $C_{\text{thr}}$ . The B5 model outperforms all other models in terms of minDCF,  $C_{\text{thr}}$ , and EER at 0.043, 0.235, and 1.5%, respectively. Owing to its outstanding performance, the B5 model has been selected as the augmented model for further experimental exploration.

Although current experimental results do not conclusively determine which of the three different DA policies is most effective. We recommend prioritizing experimental exploration under random-DA and cascade-DA policies for speech spoofing detection tasks.

### 4.3. Effect of GAM-based Co-enhancement Strategy

With the B5 model’s good results, we also investigate whether the spoofing detection performance can be further improved by using the GAM method. Table 4 shows the performance of various DA methods and GAM method on the Track 1 progress phase.

For the effect of data augmentation, the C1 and C2 models did not significantly improve minDCF and EER over the B5 model in the progress phase. Specifically, the B5 model using RIR-TimeMask (C8 and C9 models) and its combination with the Mixup (C10 and C11 models) outperformed the C2 and C3 models across most metrics, indicating that more complex augmentation can be learning more robust features. In addition, the comparison among models from C8 to C11 shows that the Mixup method significantly improves both minDCF and EER, which suggests that it contributes to the improvement’s generalizability. The GAM method, particularly in B5 and C9 models, improved minDCF and EER, effectively enhancing model generalizability. The experimental results demonstrate the importance of selecting appropriate data augmentation and optimization techniques to enhance spoofing detection performance.



Table 3: Effect of A9 model with various DA methods on Track 1 progress phase. The highlighted model was selected for further fine-tuning to enhance its generalizability.

Model ID	DA Policy	DA Method	Optimizer	Sample Points of Training	Sample Points of Progress	minDCF	actDCF	$C_{ur}$	EER
B1	Single	Amplitude	Adam	64,600	64,600	0.137	0.322	0.466	6.76
B2	Random	a-law or $\mu$ -law		64,600	64,600	0.063	<b>0.108</b>	0.287	2.32
B3	Random	Mix		64,600	64,600	0.139	0.454	0.553	6.21
B4	Cascade	RIR-TimeMask		64,600	64,600	0.057	0.420	1.508	2.05
B5	Cascade	RIR-TimeMask		96,000	96,000	<b>0.043</b>	<b>0.116</b>	<b>0.235</b>	<b>1.50</b>
B6	Cascade	RIR-TimeMask		128,000	128,000	0.067	0.143	0.302	2.46

Table 4: Effect of the B5 model under GAM-based co-enhancement strategy on Track 1 progress phase.

Model ID	DA Policy	DA Method	Optimizer	Sample Points of <i>Training</i>	Sample Points of <i>Progress</i>	minDCF	actDCF	$C_{ur}$	EER
C1	Single	RIR	Adam + GAM	64,600	96,000	0.058	<b>0.062</b>	0.111	2.07
C2	Random	a-law or $\mu$ -law			96,000	0.046	0.067	0.204	1.63
C3	Random	Mix			64,600	0.064	0.322	0.661	2.26
C4					96,000	0.050	0.194	0.365	1.79
C5	Cascade	LnL-ISD			64,600	0.057	0.230	0.367	2.06
C6					96,000	0.048	0.155	0.257	1.71
C7	Cascade	RIR + Mix			96,000	0.046	0.149	0.221	1.63
C8	Cascade	RIR-TimeMask			64,600	0.051	0.189	0.314	1.84
C9					96,000	0.041	0.190	0.276	1.48
C10	Cascade	RIR-TimeMask + Mixup			64,600	0.050	0.922	1.688	1.77
C11					96,000	0.038	0.840	1.334	1.39
C12	Cascade	RIR + Noise-Filter			96,000	<b>0.035</b>	0.087	<b>0.108</b>	<b>1.30</b>

Table 5: The performance of different fused systems was evaluated on the ASVspoof 5 Track 1 database. “Prog.” and “Eval.” refer to the progress and evaluation sets, respectively.

Phase	ID	System	minDCF	actDCF	$C_{ur}$	EER
Prog.	D1	B4 + B5	0.039	0.307	0.635	1.33
	D2	B1 ~ B6	0.037	<b>0.167</b>	<b>0.305</b>	1.31
	D3	C8 + C9	0.040	0.633	0.456	1.41
	D4	C11 + C12	<b>0.027</b>	0.269	0.366	<b>0.99</b>
Eval.	D4	C11 + C12	0.115	0.573	0.956	4.04

#### 4.4. Comparison Analysis of Different Fused Systems

Table 5 shows the performance of the four fused systems on either the *progress* or *evaluation* sets of ASVspoof 5. We observed that score-level average fusion enhances model performance compared to individual detection models, particularly in minDCF and EER metrics. Fusing C11 and C12 models (D4) resulted in optimal progress phase performance, achieving a minDCF of 0.027 and an EER of 0.99%. However, the D4 system exhibited a significant performance discrepancy between the progress and evaluation phases, highlighting the challenging nature of the *evaluation* set.

#### 4.5. Impact of Different Sample Points

Table 3 also shows a comparison in terms of sample points for the model training. Using 96,000 sample points of input speech, the B5 model achieved a lower actDCF and  $C_{ur}$  than B4 and B6. The B6 model exhibited poor performance, indicating that increasing the number of training samples does not necessarily

enhance the model’s detection capabilities. In fact, inputting more sample points for training may reduce the model’s generalization ability. Optimizing training with an appropriate number of sample points is more beneficial for improving detection performance than simply increasing the amount of training data.

The results presented in Table 4 reveal that when the model was trained using input speech with 64,600 sample points, a significant performance improvement was observed during the inference stage when utilizing input speech with 96,000 sample points. This phenomenon may be associated with the different utterance duration distribution in the *progress* set. More studies are required to verify this relationship further and analyze it.

## 5. Conclusion

This paper describes the SZU-AFS system for Track 1 of the ASVspoof 5 Challenge under open conditions. Instead of focusing on various pre-trained feature fusion and complex score fusion methods, we used DA and GAM enhancement strategies to improve spoofing detection generalization. The final best fused system submitted achieved 0.115 minDCF and 4.04% EER on the ASVspoof 5 challenge *evaluation* set.

The experiments produced a few valuable findings. First, applying the RIR-TimeMask method for data augmentation has proven more effective. Building on this, employing a cascade-DA strategy can further improve model performance. Second, the GAM method significantly improves model generalization when combined with the Adam optimizer on both *progress* and *evaluation* sets despite the lengthy training time required. Due to time constraints, the model was fine-tuned in two stages. Using the GAM method throughout the entire process might have produced better results.

## 6. Acknowledgments

We would like to thank the organizers for hosting the ASVspoof 5 Challenge. This work was supported in part by NSFC (Grant U23B2022, U22B2047) and Guangdong Provincial Key Laboratory (Grant 2023B1212060076).

## 7. References

- [1] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li, “Spoofing and countermeasures for speaker verification: A survey,” *Speech Communication*, vol. 66, pp. 130–153, 2015.
- [2] Zhizheng Wu, Tomi Kinnunen, Nicholas W. D. Evans, Junichi Yamagishi, Cemal Hanilçi, Md. Sahidullah, and Aleksandr Sizov, “ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge,” in *Proc. Interspeech*, 2015, pp. 2037–2041.
- [3] Tomi Kinnunen, Md. Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas W. D. Evans, Junichi Yamagishi, and Kong-Aik Lee, “The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection,” in *Proc. Interspeech*, 2017, pp. 2–6.
- [4] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas W. D. Evans, Tomi H. Kinnunen, and Kong Aik Lee, “ASVspoof 2019: Future horizons in spoofed and fake audio detection,” in *Proc. Interspeech*, 2019, pp. 1008–1012.
- [5] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md. Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al., “ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection,” in *Proc. ASVspoof Challenge Workshop*, 2021, pp. 47–54.
- [6] Xin Wang, Héctor Delgado, Hemlata Tak, Jee-weon Jung, Hye-jin Shim, Massimiliano Todisco, Ivan Kukanov, Xuechen Liu, Md. Sahidullah, Tomi Kinnunen, Nicholas Evans, Kong Aik Lee, and Junichi Yamagishi, “ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale,” in *ASVspoof Workshop 2024 (accepted)*, 2024.
- [7] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury, “Generalization of audio deepfake detection,” in *Proc. Odyssey*, 2020, pp. 132–137.
- [8] João Monteiro, Jahangir Alam, and Tiago H. Falk, “Generalized end-to-end detection of spoofing attacks to automatic speaker recognizers,” *Computer Speech & Language*, vol. 63, pp. 101096, 2020.
- [9] Xin Wang and Junichi Yamagishi, “A comparative study on recent neural spoofing countermeasures for synthetic speech detection,” in *Proc. Interspeech*, 2021, pp. 4259–4263.
- [10] You Zhang, Fei Jiang, and Zhiyao Duan, “One-class learning towards synthetic voice spoofing detection,” *IEEE Signal Processing Letters*, vol. 28, pp. 937–941, 2021.
- [11] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert, “MLS: A large-scale multilingual dataset for speech research,” in *Proc. Interspeech*, 2020, pp. 2757–2761.
- [12] Seyed Omid Sadjadi, Craig S Greenberg, Elliot Singer, Douglas A Reynolds, and Lisa Mason, “NIST 2020 CTS speaker recognition challenge evaluation plan,” 2020.
- [13] Hye-jin Shim, Jee-weon Jung, Tomi Kinnunen, Nicholas W. D. Evans, Jean-François Bonastre, and Itshak Lapidot, “a-DCF: an architecture agnostic metric with application to spoofing-robust speaker verification,” in *Proc. Odyssey*, 2024, pp. 158–164.
- [14] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NIPS*, 2020, vol. 33, pp. 12449–12460.
- [15] Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas W. D. Evans, “AASIST: audio anti-spoofing using integrated spectro-temporal graph attention networks,” in *Proc. ICASSP*, 2022, pp. 6367–6371.
- [16] Juan M. Martín-Doñas and Aitor Álvarez, “The vi-comtech audio deepfake detection system based on wav2vec2 for the 2022 ADD challenge,” in *Proc. ICASSP*, 2022, pp. 9241–9245.
- [17] Jin Woo Lee, Eungbeom Kim, Junghyun Koo, and Kyogu Lee, “Representation selective self-distillation and wav2vec 2.0 feature exploration for spoof-aware speaker verification,” in *Proc. Interspeech*, 2022, pp. 2898–2902.
- [18] Jiafeng Zhong, Bin Li, and Jiangyan Yi, “Enhancing partially spoofed audio localization with boundary-aware attention mechanism,” *arXiv preprint arXiv:2407.21611*, 2024.
- [19] Yujie Yang, Haochen Qin, Hang Zhou, Chengcheng Wang, Tianyu Guo, Kai Han, and Yunhe Wang, “A robust audio deepfake detection system via multi-view feature,” in *Proc. ICASSP*, 2024, pp. 13131–13135.
- [20] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xi-angzhan Yu, and Furu Wei, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [21] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [22] Xin Wang and Junichi Yamagishi, “Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders,” in *Proc. ICASSP*, 2023, pp. 1–5.
- [23] Eros Rosello, Alejandro Gómez Alanís, Angel M. Gomez, and Antonio M. Peinado, “A conformer-based classifier for variable-length utterance processing in anti-spoofing,” in *Proc. Interspeech*, 2023, pp. 5281–5285.
- [24] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee-weon Jung, Junichi Yamagishi, and Nicholas W. D. Evans, “Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation,” in *Proc. Odyssey*, 2022, pp. 112–119.

- [25] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao, “Audio deepfake detection: A survey,” *arXiv preprint arXiv:2308.14970*, 2023.
- [26] Yuxiong Xu, Bin Li, Shunquan Tan, and Jiwu Huang, “Research progress on speech deepfake and its detection techniques,” *Journal of Image and Graphics*, vol. 29, no. 08, pp. 2236–2268, 2024.
- [27] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher, “End-to-end anti-spoofing with rawnet2,” in *Proc. ICASSP*, 2021, pp. 6369–6373.
- [28] Hemlata Tak, Madhu R. Kamble, Jose Patino, Massimiliano Todisco, and Nicholas W. D. Evans, “Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing,” in *Proc. ICASSP*, 2022, pp. 6382–6386.
- [29] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz, “Mixup: Beyond empirical risk minimization,” in *Proc. ICLR*, 2018.
- [30] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” in *Proc. ICLR*, 2021.
- [31] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi, “ASAM: adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks,” in *Proc. ICML*, 2021, vol. 139, pp. 5905–5914.
- [32] Hye-jin Shim, Jee-weon Jung, and Tomi Kinnunen, “Multi-dataset co-training with sharpness-aware optimization for audio anti-spoofing,” in *Proc. Interspeech*, 2023, pp. 3804–3808.
- [33] Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui, “Gradient norm aware minimization seeks first-order flatness and improves generalization,” in *Proc. CVPR*, 2023, pp. 20247–20257.
- [34] Niko Brümmer and Johan A. du Preez, “Application-independent evaluation of speaker detection,” *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.
- [35] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015.