

# PadAug: Robust Speaker Verification with Simple Waveform-Level Silence Padding

Zijun Huang<sup>1,3</sup>, Chengdong Liang<sup>2</sup>, Jiadi Yao<sup>1,3</sup>, and Xiao-Lei Zhang<sup>1(✉),3</sup>

<sup>1</sup> School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China

xiaolei.zhang@nwpu.edu.cn

<sup>2</sup> Guasemi, Shanghai, China

<sup>3</sup> Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen, China

**Abstract.** The presence of non-speech segments in utterances often leads to the performance degradation of speaker verification. Existing systems usually use voice activation detection as a preprocessing step to cut off long silence segments. However, short silence segments, particularly those between speech segments, still remain a problem for speaker verification. To address this issue, in this paper, we propose a simple wave-level data augmentation method, *PadAug*, which aims to enhance the system's robustness to silence segments. The core idea of *PadAug* is to concatenate silence segments with speech segments at the waveform level for model training. Due to its simplicity, it can be directly applied to the current state-of-the-art architectures. Experimental results demonstrate the effectiveness of the proposed *PadAug*. For example, applying *PadAug* to ResNet34 achieves a relative equal error rate reduction of 5.0% on the voxceleb dataset. Moreover, the *PadAug* based systems are robust to different lengths and proportions of silence segments in the test data.

**Keywords:** Speaker verification · Data augmentation · Silence padding.

## 1 Introduction

Speaker Verification (SV) aims at determining whether a speech utterance is uttered from an enrolled speaker or not. A conversational SV system mainly consist of two parts: an embedding vector extractor [4] which produces discriminative embeddings for utterances of different speakers, and a back-end classifier, e.g. PLDA [10] or Cosine similarity scoring, which computes the similarity score between the testing utterances and enrollment utterances. Currently, deep-based SV systems [1, 15, 21] have achieved the state-of-the-art performance in many scenarios such as intelligent housing systems [28], adversarial attack [27], anti-spoofing [2, 30]. However, their performance were still affected strongly by silence segments [23, 9], since that the acoustic features of silence segments behave like additive noise of speech when generating speaker embeddings via e.g. x-vectors.

A common way of dealing with the silence segments is to use a voice activation detection (VAD) [22] based front-end to filter out silence segments. VAD usually uses a statistical model to model the characteristics of speech and non-speech signals such as energy, and then uses a frame dropping strategy to detect long silence segments. However, to avoid dropping unvoiced speech e.g. consonants, VAD always uses hang-before and hangover criteria [24] to keep short periods around voiced speech, where the short periods may be silence segments that affect the robustness of SV.

To deal with the short silence periods, another common approach is the attentive pooling, e.g. [6], which allocates high weights to the frames that are important to SV embeddings, behaves like an implicit VAD in SV. However, attentive pooling considers too much information for generating the SV embeddings, instead of merely removing the negative effect of the short silence periods.

In this paper, we resort another simple way to address the above issue—data augmentation. This research direction seems far from explored yet. To our knowledge, existing studies mostly focus on improving the robustness of speech processing systems against noise environments only. For example, [18] synthesised noisy data by superimposing clean speech and noise for speech recognition. [13] introduced speed perturbation on raw speech for speech recognition. [11] explored simulated data with an acoustic room simulator in far-field ASR systems. [17] proposed spectrum augmentation to the log mel spectrogram of data. [23] applied length penalty to improve the latency of streaming speech recognition models. [21] proposed to add additive noise and reverberation to speech for improving the robustness of SV in various noisy conditions. [12] proposed partial additive noise to enhance the robustness of SV towards background noise.

Given the aforementioned analysis, motivated by [23], this paper proposes a simple and effective data augmentation method, named silence padding augmentation (*PadAug*). *PadAug* can be simply described as that it directly concatenates silence segments with speech utterances for training. Essentially, it makes a SV model biased towards learning the discriminative information between speaker segments and non-speaker segments, thus avoiding putting too much focus on the silence segments.

Different from existing data augmentation methods of SV, which directly adds additive noise to speech, e.g. partial additive noise *PAS* [12], for enhancing the robustness of SV systems in noisy environment (see Section 2 for more related work), *PadAug* aims at enhancing the performance of SV in scenarios where a test utterance contains silence segments that appear in random positions with different lengths.

We applied *PadAug* on several popular end-to-end SV networks to verify the effectiveness of the proposed method on improving the robustness of SV against silence segments. Extensive experimental results on Voxceleb demonstrate the effectiveness of the proposed method.

## 2 Related work

### 2.1 Waveform augmentation

The data augmentation strategies in [21] were widely used to enhance the performance for noisy scenarios, where each utterance file is modified by the following approaches:

*Additive noises*: Music, Babble, and Noise files from MUSAN dataset [19] are randomly selected and added at a SNR margin. [12] extended the method to partial additive noises by controlling the duration of noise segments to further enhance the robustness of SV in noisy environments.

*Reverberation*: Simulated room impulse responses (RIRs) are used to add reverberation into the speech recordings [14]. Many papers considered using the RIR\_NOISES dataset [8] to generate the reverberant training data.

### 2.2 Spectrogram augmentation

Data augmentation on spectrograms was proposed by [17] for speech recognition. The spectrograms of speaker utterances are augmented by *time warping*, *time masking* and *frequency masking*. [23] applied *length penalty* to the spectrograms by trimming the trailing frames, leading frames and corresponding padding frames of the spectrograms.

## 3 Proposed method

Motivated by the length penalty and attention mechanism, *PadAug* aims to augment data at the waveform-levels, which is directly applied to various SV systems for enhancing the robustness of SV against silence segments.

As shown in Figure 1, *PadAug* modifies the input data of speaker by trimming the waveform and adding silence segments. The silence padding consists of Gaussian white noise and randomly concatenates the speaker segments on the head, central body and tail. Algorithm 1 gives the details of the proposed data augmentation progress.

## 4 Experiments

### 4.1 Datasets

We used the VoxCeleb dataset [16, 3] for experiments. We trained the speaker verification networks on the development set of Vox-Celeb2 [3] which consists of 1,092,009 utterances from 5,994 speakers. We used three types of evaluation sets VoxCeleb1-O, VoxCeleb1-H and VoxCeleb1-H, which are drawn from the VoxCeleb1 training set [16] for evaluation. We adopted the noise datasets from MUSAN [19] and RIRs [8] for the conventional data augmentation in the model training.

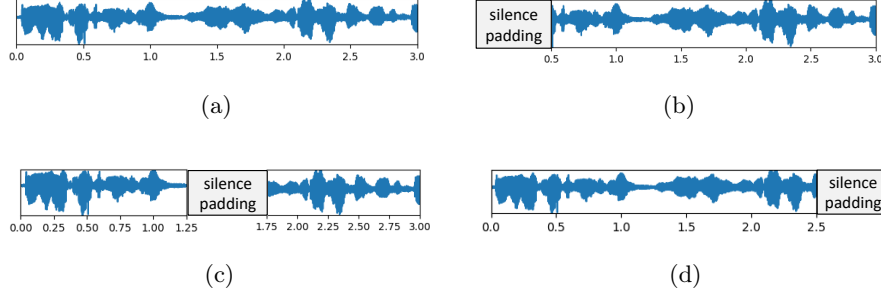


Fig. 1: Illustration of PadAug: From top to bottom: (a) Waveform of the utterance without silence padding, (b) Waveform of the utterance with silence padding on the head, (c) Waveform of the utterance with silence padding on the central body, (d) Waveform of the utterance with silence padding on the tail.

To construct the training data, each utterance of Voxceleb was clipped into a 3-second segment, where the voiced speech was guaranteed to be from 1 second to 3 seconds. To simulate the real-world scenario where an utterance is short and filled with random silence segments, we constructed the following test datasets:

- Original: The original test sets, which are VoxCeleb1-O, VoxCeleb1-H and VoxCeleb1-H respectively.
- Chunk3s: The test utterances were clipped into segments of 3 seconds long.
- Chunk3s+Head1s+Tail1s: Each 3-second utterance in Chunk3s was padded with a 1-second silence at the head and tail respectively.
- Chunk3s+Head1s+Tail1s+Mid1s: Each 3-second utterance in Chunk3s was padded with a 1-second silence on the head, middle, and tail respectively.

## 4.2 Comparison methods

For the proposed *PadAug*, our experiments contain two position strategies:

- **Head-tail based *PadAug* (PadAug(HT))**: Silence paddings were concatenated at the head and tail of each speech segment.
- **Head-mid-tail based *PadAug* (PadAug(HMT))**: Silence paddings were concatenated at the head, body and tail of each speech segment.

Note that the length of each padding component was random. We adopted the proposed augmentation method on four popular and state-of-the-art architectures: Resnet-34 [29], ECAPA-TDNN-512 [20], ECAPA-TDNN-1024 [7], and CAM++ [26]. We implemented the models by ourselves to maintain a fair comparison on the investigation of whether using the proposed method or not.

We took the no-padding strategy in Figure 1 (a) as the most important baseline, which only adds additive noise or reverberation to the input data with a probability of 0.6.

**Algorithm 1:** Applying *PadAug* to a mini-batch

---

**Input:** mini-batch  $X$ , noise  $N$ , minimum duration  $T_{min}$ , maximum duration  $T_{max}$ ,  $SNR_{min}$ ,  $SNR_{max}$ ,  $use\_mid$ .

**Output:** training mini-batch  $X'$

*/\* data is temporal sequence \*/*

```

1  $X' = [];$ 
2 for  $x$  in  $X$  do
3   Sample  $T_s \sim \text{Randint}(T_{min}, T_{max});$ 
4    $x \leftarrow \text{Random\_Chunk}(x, T_s);$ 
5    $L_{pad} \leftarrow T_{max} - T_s;$ 
6   Sample  $L_{head} \sim \text{Randint}(0, L_{pad});$ 
7   if  $use\_mid$  then
8     Sample  $L_{mid} \sim \text{Randint}(0, L_{pad} - L_{head});$ 
9      $L_{tail} \leftarrow L_{pad} - L_{head} - L_{mid};$ 
10  else
11     $L_{mid} \leftarrow 0;$ 
12     $L_{tail} \leftarrow L_{pad} - L_{head};$ 
13  end
14  Sample  $SNR \sim \text{Randint}(SNR_{min}, SNR_{max});$ 
15   $n \leftarrow \text{wgn}(x, SNR);$ 
16   $\text{Seg}_{head} \leftarrow n(0, L_{head});$ 
17   $\text{Seg}_{mid} \leftarrow n(L_{head}, L_{head} + L_{mid});$ 
18   $\text{Seg}_{tail} \leftarrow n(L_{head} + L_{mid}, L_{pad});$ 
19  Sample  $P_{mid} \sim \text{Randint}(0, T_s);$ 
20   $x_{left} \leftarrow X(0, P_{mid});$ 
21   $x_{right} \leftarrow X(P_{mid}, T_s);$ 
22   $x' \leftarrow \text{concat}(\text{Seg}_{head}, x_{left}, \text{Seg}_{mid}, x_{right}, \text{Seg}_{tail});$ 
23   $X' \leftarrow \text{append}(X', x');$ 
24 end

```

---

### 4.3 Implementation details

We used the open-source speaker embedding learning toolkit Wespeaker<sup>1</sup>[25] to implement the proposed data augmentation method and baselines. Speed perturbation was conducted by randomly changing the speed of the utterances with a ratio of 0.9 or 1.1. We adopted 80-dimensional log Mel-filter banks (Fbank) which were pre-emphasized by a Hamming window with a width 25ms and a window shift of 10ms as the input features. All training segments were chunked into 300 frames and each was normalized by the cepstral mean normalization (CMN). The loss functions of all models were the angular additive margin softmax (AAM-Softmax) [5], where the scale was set to 32, and the initial margin was set to 0 and updated until 0.2 by the margin schedule [25]. The initial learning rate was set to  $10^{-1}$ . We used the warm-up and exponential decrease schedule in [25] to update the learning rate until  $5 \times 10^{-5}$ .

<sup>1</sup> <https://github.com/wenet-e2e/wespeaker>

Table 1: Comparison between the SV models with the proposed *PadAug(HT)* and those without the proposed method.

SV Model	Augmentation method	Test set	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
			EER	minDCF	EER	minDCF	EER	minDCF
ResNet34	<del>X</del>	Original	0.962	0.091	1.007	0.110	1.859	0.178
	PadAug(HT)	Original	0.914	0.077	0.978	0.104	1.798	0.174
	<del>X</del>	Chunk3s	1.345	0.143	1.259	0.142	2.348	0.227
	PadAug(HT)	Chunk3s	1.170	0.114	1.261	0.140	2.326	0.231
	<del>X</del>	Chunk3s+Head1s+Tail1s	1.409	0.142	1.415	0.156	2.560	0.249
	PadAug(HT)	Chunk3s+Head1s+Tail1s	1.170	0.122	1.261	0.144	2.311	0.229
ECAPA-TDNN-512	<del>X</del>	Original	1.090	0.115	1.227	0.134	2.278	0.219
	PadAug(HT)	Original	1.064	0.098	1.174	0.129	2.177	0.207
	<del>X</del>	Chunk3s	1.468	0.162	1.526	0.176	2.831	0.275
	PadAug(HT)	Chunk3s	1.526	0.180	1.557	0.172	2.828	0.276
	<del>X</del>	Chunk3s+Head1s+Tail1s	1.781	0.208	1.864	0.205	3.399	0.319
	PadAug(HT)	Chunk3s+Head1s+Tail1s	1.435	0.162	1.569	0.174	2.831	0.278
ECAPA-TDNN-1024	<del>X</del>	Original	0.909	0.083	1.056	0.113	1.942	0.189
	PadAug(HT)	Original	0.830	0.088	0.995	0.108	1.845	0.183
	<del>X</del>	Chunk3s	1.175	0.121	1.313	0.147	2.461	0.248
	PadAug(HT)	Chunk3s	1.165	0.135	1.316	0.150	2.447	0.242
	<del>X</del>	Chunk3s+Head1s+Tail1s	1.473	0.159	1.547	0.175	2.899	0.280
	PadAug(HT)	Chunk3s+Head1s+Tail1s	1.154	0.144	1.327	0.152	2.411	0.241
CAM++	<del>X</del>	Original	0.723	0.122	0.940	0.109	1.916	0.194
	PadAug(HT)	Original	0.707	0.097	0.911	0.106	1.818	0.178
	<del>X</del>	Chunk3s	1.042	0.155	1.187	0.143	2.388	0.240
	PadAug(HT)	Chunk3s	0.963	0.147	1.156	0.136	2.302	0.238
	<del>X</del>	Chunk3s+Head1s+Tail1s	1.249	0.184	1.387	0.171	2.767	0.281
	PadAug(HT)	Chunk3s+Head1s+Tail1s	1.000	0.150	1.172	0.138	2.281	0.232

#### 4.4 Evaluation criterion

In the test stage, we adopted cosine similarity as the scoring function without any score normalization. We employed the minimum detection cost function (minDCF) with  $P_{target} = 0.01$  and  $C_{miss} = C_{fa} = 1$  and the standard equal error rate (EER) as the evaluation protocols.

#### 4.5 Main results

Table 1 lists the main experimental results of the proposed *PadAug* and baseline on four state-of-the-art SV models. From the table, we see that the proposed *PadAug* improve the system’s performance in all cases. For example, for the ‘Chunk3s’ test set of Voxceleb1-O, the ResNet34 model with *PadAug* achieves a relative EER reduction of 5.0% over that without *PadAug*. Similarly, the relative EER reduction with ECAPA-TDNN-512, ECAPA-TDNN-1024, and CAM++ are 2.4%, 8.7%, and 2.2% respectively.

We also can see that, when the silence period increases, the performance of the SV systems gets worse rapidly, and *PadAug* can effectively reduce the performance degradation. Taking the experimental results on the ‘*Chunk3s+Head1s+Tail1s*’ test set of Voxceleb1-O as an example. The ResNet34 model with *PadAug* achieves a relative EER reduction of 17.0% over that without *PadAug*. Similarly,

Table 2: Comparison between *PadAug(HT)* and *PadAug(HMT)*, where the SV model is ResNet34.

Method	Testset type	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
		EER	minDCF	EER	minDCF	EER	minDCF
no <i>PadAug</i>	Original	0.962	0.091	1.007	0.110	1.859	0.178
PadAug(HT)	Original	0.914	<b>0.077</b>	<b>0.978</b>	<b>0.104</b>	<b>1.798</b>	0.174
PadAug(HMT)	Original	<b>0.904</b>	<b>0.077</b>	0.990	0.107	1.809	<b>0.173</b>
no <i>PadAug</i>	Chunk3s	1.345	0.143	<b>1.259</b>	0.142	2.348	<b>0.227</b>
PadAug(HT)	Chunk3s	<b>1.170</b>	<b>0.114</b>	1.261	<b>0.140</b>	<b>2.326</b>	0.231
PadAug(HMT)	Chunk3s	1.233	0.125	1.278	0.144	2.345	0.234
no <i>PadAug</i>	Chunk3s+Head1s+Tail1s	1.409	0.142	1.415	0.156	2.560	0.249
PadAug(HT)	Chunk3s+Head1s+Tail1s	<b>1.170</b>	0.122	<b>1.261</b>	0.144	<b>2.311</b>	<b>0.229</b>
PadAug(HMT)	Chunk3s+Head1s+Tail1s	1.207	<b>0.120</b>	1.269	<b>0.142</b>	2.328	0.230
no <i>PadAug</i>	Chunk3s+Head1s+Tail1s+Mid1s	1.526	0.169	1.512	0.170	2.724	0.265
PadAug(HT)	Chunk3s+Head1s+Tail1s+Mid1s	<b>1.213</b>	<b>0.133</b>	1.309	0.147	2.365	<b>0.227</b>
PadAug(HMT)	Chunk3s+Head1s+Tail1s+Mid1s	1.223	0.146	<b>1.302</b>	<b>0.144</b>	<b>2.363</b>	0.234

Table 3: Comparison between VAD and the proposed *PadAug(HT)* on the 'chunk3s+head1s+tail1s' test sets.

Model	Approch	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
		EER	minDCF	EER	minDCF	EER	minDCF
ResNet34	VAD	1.292	0.135	1.300	0.148	2.358	0.234
	PadAug(HT)	1.170	0.122	1.261	0.144	2.311	0.229
ECAPA-TDNN-512	VAD	1.431	0.169	1.577	0.178	2.926	0.285
	PadAug(HT)	1.435	0.162	1.569	0.174	2.831	0.278
ECAPA-TDNN-1024	VAD	1.223	0.142	1.345	0.155	2.515	0.249
	PadAug(HT)	1.154	0.144	1.327	0.152	2.411	0.241
CAM++	VAD	1.026	0.149	1.219	0.151	2.444	0.250
	PadAug(HT)	1.000	0.150	1.172	0.138	2.281	0.232

the relative EER reduction with ECAPA-TDNN-512, ECAPA-TDNN-1024, and CAM++ are 19.4%, 21.7%, and 19.9% respectively.

Table 2 lists the results of the ResNet34 model with the two variants of *PadAug*, i.e. *PadAug(HT)* and *PadAug(HMT)*. From the table, we see that the two variants perform quite similarly, and both outperforms the method without *PadAug*. Note that the performance on the other three representative SV models is similar with that on ResNet34. Due to the length limitation, we omit the result here.

#### 4.6 Robustness to the length of silence period

To demonstrate that *PadAug* the advantage of SV is insensitive to the length of silence period in the test data, we controlled the silence-to-speech ratio in a set of  $[0/3, 1/3, \dots, 8/3]$ , where the ratio of e.g. '2/3' denotes that a 3-second chunk was padded with a 2-second silence period.

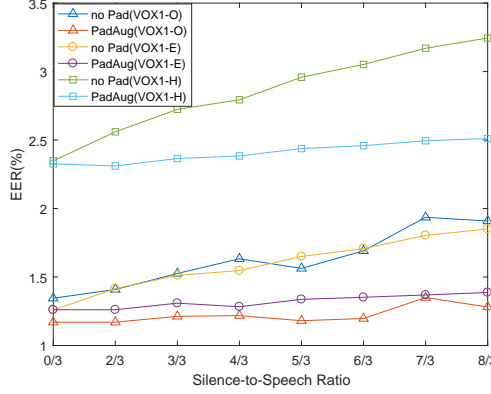


Fig. 2: EER (%) of the proposed method with respect to silence-to-speech ratio.

Figure 2 shows the results of the proposed method with respect to the silence-to-speech ratio. From the figure, we can see that, when the silence-to-speech ratio increases, the EER curves of the *PadAug*-based systems are stable, while the EER of the systems without *PadAug* increases rapidly.

#### 4.7 Comparison to VAD

In order to show that the proposed *PadAug* overcomes the shortcomings of the VAD-based method, we compared the SV systems that used VAD instead of the proposed *PadAug* with the SV systems that used *PadAug* without VAD. We implemented VAD by webrtcVAD<sup>2</sup>, which is a widely-used Gaussian mixture model based VAD.

Table 3 lists the results of comparison on the 'Chunk3s+Head1s+Tail1s' test sets. From the table, we see that the proposed method outperforms or at least does not perform worse than the VAD-based methods. Taking the results on ResNet34 as an example, the proposed *PadAug* outperforms VAD with a relative EER reduction of 9.4% on the Voxceleb1-O dataset.

#### 4.8 Comparison to attentive pooling

From the aforementioned experiments, we see that the proposed method improve the performance of the attentive pooling based SV systems. However, it is known that the attentive pooling itself behaves like an implicit VAD in SV. To compare the proposed *PadAug* with the attentive pooling directly, we firstly used *temporal statistics pooling (TSP)*[21] as the pooling layer of the ResNet34 SV model, where the implicit VAD function has been removed. Then, we applied the proposed *PadAug* to the TSP-based ResNet34 SV model.

<sup>2</sup> <https://github.com/wiseman/py-webrtcvad>



Table 4: Comparison between attentive statistics pooling (ASP) and the temporal statistics pooling (TSP) with PadAug(HT), where the SV model is ResNet34. ‘C+H+T’ is short for ‘Chunk3s+Head1s+Tail1s’. ‘C+H+T+M’ is short for ‘Chunk3s+Head1s+Tail1s+Mid1s’.

Method	Test set	Voxceleb1-O		Voxceleb1-E		Voxceleb1-H	
		EER	minDCF	EER	minDCF	EER	minDCF
ASP (no PadAug)	C+H+T	1.409	0.142	1.415	0.156	2.560	0.249
TSP+PadAug(HT)	C+H+T	1.213	0.127	1.322	0.151	2.460	0.235
ASP (no PadAug)	C+H+T+M	1.526	0.169	1.512	0.170	2.724	0.265
TSP+PadAug(HT)	C+H+T+M	1.287	0.143	1.344	0.158	2.498	0.244

Table 4 lists the comparison result. From the result, we see clearly that the proposed *PadAug* plus TSP clearly outperforms the attentive statistics pooling (ASP) without *PadAug* in all cases, which indicates that *PadAug* is better than ASP in dealing with short silence periods.

## 5 Conclusions

This paper proposed *PadAug* a novel wave-level data augmentation method, to handle the performance degradation of SV system caused by short silence segments that are not easily removable by e.g. VAD. It can be easily described as that the training data is concatenated with short silence segments in the wave level directly. Extensive experimental results on the Voxceleb dataset demonstrate the effectiveness of the proposed method.

## References

1. Bai, Z., Zhang, X.L.: Speaker recognition based on deep learning: An overview. *Neural Networks* **140**, 65–99 (2021)
2. Chen, X., Yao, J., Zhang, X.L.: Masking speech feature to detect adversarial examples for speaker verification. In: 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). pp. 191–195. IEEE (2022)
3. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: Deep speaker recognition. arXiv preprint arXiv:1806.05622 (2018)
4. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing* **19**(4), 788–798 (2010)
5. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
6. Desplanques, B., Thienpondt, J., Demuynck, K.: Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. arXiv preprint arXiv:2005.07143 (2020)

7. Garcia-Romero, D., McCree, A., Snyder, D., Sell, G.: Jhu-hltcoe system for the voxsrc speaker recognition challenge. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7559–7563. IEEE (2020)
8. Habets, E.A.: Room impulse response generator. Technische Universiteit Eindhoven, Tech. Rep **2**(2.4), 1 (2006)
9. Hasan, T., Saeidi, R., Hansen, J.H., Van Leeuwen, D.A.: Duration mismatch compensation for i-vector based speaker recognition systems. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 7663–7667. IEEE (2013)
10. Ioffe, S.: Probabilistic linear discriminant analysis. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part IV 9. pp. 531–542. Springer (2006)
11. Kim, C., Misra, A., Chin, K.K., Hughes, T., Narayanan, A., Sainath, T.N., Bacchiani, M.: Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. In: Interspeech. pp. 379–383 (2017)
12. Kim, W., Shin, H.s., Kim, J.h., Heo, J., Lim, C.y., Yu, H.J.: Pas: Partial additive speech data augmentation method for noise robust speaker verification. arXiv preprint arXiv:2307.10628 (2023)
13. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. Interspeech 2015 (2015)
14. Ko, T., Peddinti, V., Povey, D., Seltzer, M.L., Khudanpur, S.: A study on data augmentation of reverberant speech for robust speech recognition. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5220–5224. IEEE (2017)
15. Li, C., Ma, X., Jiang, B., Li, X., Zhang, X., Liu, X., Cao, Y., Kannan, A., Zhu, Z.: Deep speaker: an end-to-end neural speaker embedding system. arXiv preprint arXiv:1705.02304 (2017)
16. Nagrani, A., Chung, J.S., Zisserman, A.: Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612 (2017)
17. Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V.: SpecAugment: A simple data augmentation method for automatic speech recognition. Interspeech 2019 (2019)
18. Seltzer, M.L., Yu, D., Wang, Y.: An investigation of deep neural networks for noise robust speech recognition. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 7398–7402. IEEE (2013)
19. Snyder, D., Chen, G., Povey, D.: Musan: A music, speech, and noise corpus. arXiv preprint arXiv:1510.08484 (2015)
20. Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D., Khudanpur, S.: Speaker recognition for multi-speaker conversations using x-vectors. In: ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP). pp. 5796–5800. IEEE (2019)
21. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust dnn embeddings for speaker recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5329–5333. IEEE (2018)
22. Sohn, J., Kim, N.S., Sung, W.: A statistical model-based voice activity detection. IEEE signal processing letters **6**(1), 1–3 (1999)

23. Song, X., Wu, D., Wu, Z., Zhang, B., Zhang, Y., Peng, Z., Li, W., Pan, F., Zhu, C.: Trintail: Low-latency streaming asr with simple but effective spectrogram-level length penalty. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
24. Vlaj, D., Kos, M., Grasic, M., Kacic, Z.: Influence of hangover and hangbefore criteria on automatic speech recognition. In: 2009 16th International Conference on Systems, Signals and Image Processing. pp. 1–4. IEEE (2009)
25. Wang, H., Liang, C., Wang, S., Chen, Z., Zhang, B., Xiang, X., Deng, Y., Qian, Y.: Wespeaker: A research and production oriented speaker embedding learning toolkit. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
26. Wang, H., Zheng, S., Chen, Y., Cheng, L., Chen, Q.: Cam++: A fast and efficient network for speaker verification using context-aware masking. arXiv preprint arXiv:2303.00332 (2023)
27. Yao, J., Chen, X., Zhang, X.L., Zhang, W.Q., Yang, K.: Symmetric saliency-based adversarial attack to speaker identification. IEEE Signal Processing Letters **30**, 1–5 (2023)
28. Yao, J., Liang, C., Peng, Z., Zhang, B., Zhang, X.L.: Branch-ecapa-tdnn: A parallel branch architecture to capture local and global features for speaker verification. In: Proc. Interspeech. pp. 1943–1947 (2023)
29. Zeinali, H., Wang, S., Silnova, A., Matějka, P., Plchot, O.: But system description to voxceleb speaker recognition challenge 2019. arXiv preprint arXiv:1910.12592 (2019)
30. Zhang, M., Xu, K., Li, H., Wang, L., Fang, C., Shi, J.: Doubleddeceiver: Deceiving the speaker verification system protected by spoofing countermeasures