Fusing Blocked Deep Features of Pre-Trained Models for Short-Duration Speaker Verification

Zhijun Zhang, Wu Guo, Jie Zhang, Yu Guan
University of Science and Technology of China
E-mail: zhangzhijun@mail.ustc.edu.cn, guowu@ustc.edu.cn, jzhang6@ustc.edu.cn, gypursue@mail.ustc.edu.cn

***Abstract*—In this paper, we propose a method to jointly use multi-layer features from pre-trained model (PTM) and hand-crafted features for short-duration speaker verification. We evenly divide the Transformer layers of PTM into shallow and deep parts, each of which are processed separately. For the shallow part, we partition feature-map of each layer into several sub-blocks along channel dimension and calibrate them via learnable weights. The weighted features from both shallow and deep parts and the hand-crafted features are then fused using attention-based fusion modules. Experimental results on the VoxCeleb datasets show the promising performance of the proposed method with an EER of 0.984% given 3s-duration trails on VoxCeleb-O.**

I. Introduction

Speaker verification (SV) aims to determine if an utterance belongs to a claimed speaker, which has a broad range of applications to e.g., biometric authentication [1], personalized services [2], forensic scenarios [3]. In the past decades, deep neural network (DNN) [4] (i.e., x-vector) has become the mainstream for SV. Various models, such as time-delay neural network (TDNN) [5], ResNet [6], ECAPA [7] and their variants[8]–[11], have been proposed and achieved remarkable results. However, it was shown that performance significantly deteriorates as the duration of input speech signal used for verification is short [12]. This is mainly due to the lack of data in modeling the speaker's voice characteristics. While from the perspective of real-world deployment, short-duration SV is more common and preferred [13].

Recently, pre-trained models (PTMs) such as Wav2vec2.0[14], HuBERT [15] and WavLM [16], have been proposed to extract general features, which can be applied to many downstream tasks including SV. PTMs are trained by self-supervised learning on large-scale unlabeled data, They are thus ideal front-end feature extractor for short-duration SV. Most PTMs contain 12/24 transformer layers, and different layers of PTMs capture semantic information or speaker identity at varying levels [17]. Intuitively, fusing the representations from all layers can improve the performance of SV. In [18], Chen *et al.* weighted the outputs of all hidden layers to fully extract speaker-related representations. Peng *et al.* proposed a multi-level fusion strategy to further exploit the complementary information of PTM and hand-crafted features for SV [19]. To take into account the gains of different layers for SV, aforementioned methods calibrate the representation of a layer through a learnable weight before fusion. However, for short duration SV, data scarcity will make some representations less discriminative. Classic fusion approaches may fail to account for the discrepancies among the representations of transformer layers, and efforts should therefore be made to fill this gap.

In this work, we use the blocked deep features of PTM to obtain more-refined deep features for short-duration SV. It is known that the representations from low layers focus on local details, resembling traditional hand-crafted features [20], while deep layers consider broader context information, capturing global semantics [17]. Based on this observation, we evenly divide the transformer layers of a PTM into shallow and deep parts, each of which are processed separately. For the shallow part, we partition the feature map from a transformer layer into sub-blocks along channel dimension and calibrate them using block-dependent weights instead of a layer-shared one. Note that the number of sub-blocks in each layer varies. The bottom layers have more sub-blocks (e.g., six in experiments), and the top layers gradually decrease. The weighted blocked features from shallow part and that from deep part are first fused using an attention-based fusion module (AFM), and they are further fused with traditional hand-crafted features for back-end modeling.

Experiments are conducted on VoxCeleb datasets using three PTMs (Wav2Vec2.0, Hubert, WavLM) as the front-end feature extractor. Results show that our method can outperform state-of-the-art (SOTA) approaches in the short-duration case. When trials are cropped to 3 seconds, it can achieve EER (equal error rate) reductions of 14.2%, 11.5% and 20.6% using the three PTMs, respectively.

The rest of the paper is organized as follows. Section II summarizes related works. Section III presents the proposed method in detail. Experimental setup and results are given in Section IV. Finally, Section VI concludes this work.

II. Related Work

A. Baseline

To provide a comprehensive understanding, we begin by elaborating on the network architecture of the baseline in this section.

The baseline follows the model in [18], which contains a PTM as the front-end feature extractor cascaded with an ECAPA-TDNN model as back-end classifier. To effectively use the complementary information of the PTM, we perform a weighted average on all layer-wise representations, i.e., replacing the use of only the output from the last transformer

979-8-3315-7206-8/25/$31.00

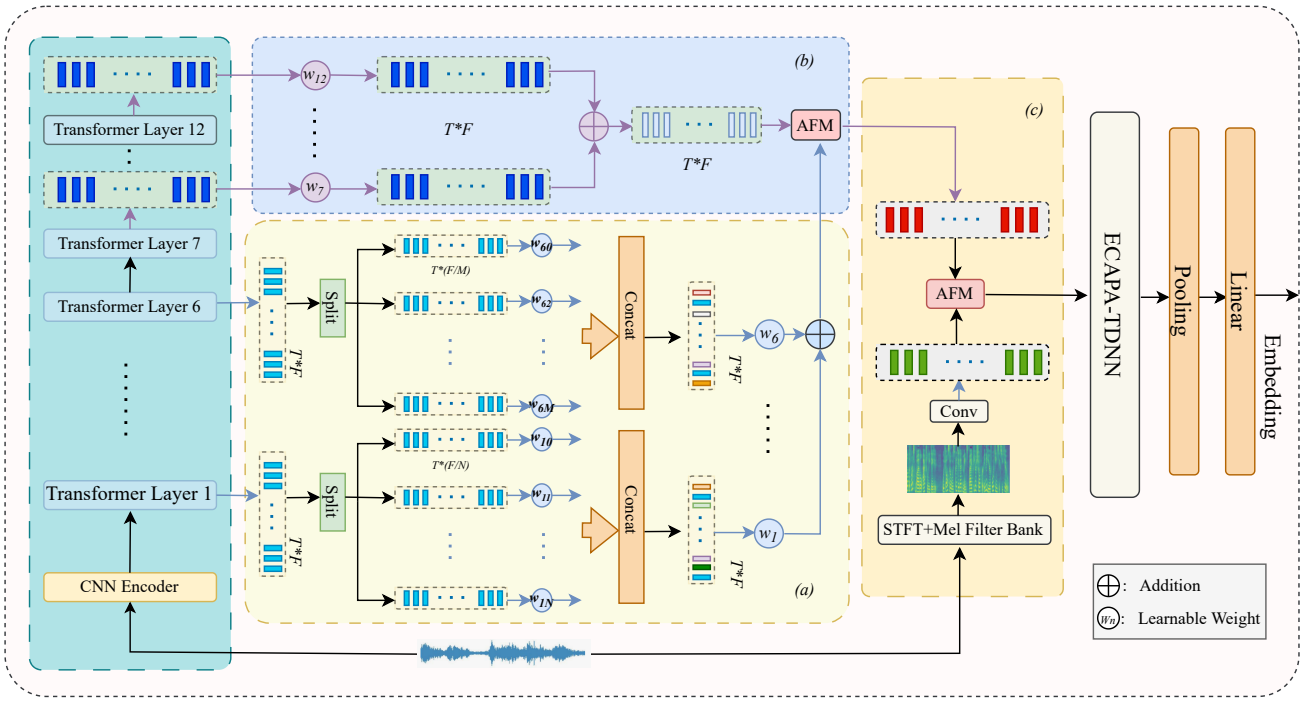

Fig. 1. Proposed framework: (a) feature partitioning and calibration in shallow part, (b) feature fusion of shallow and deep parts and (c) fusion with FBank.

layer. The weighted features are then fed into the back-end classifier to extract speaker embedding E. This process can be represented as

$$\{h^{(0)}, \ldots, h^{(l)}\} = \text{PTM}(X), \tag{1}$$

$$E = \text{ECAPA-TDNN}\left(\text{Linear}\left(\sum_{l=0}^{L} w_l \cdot h^{(l)}\right)\right), \tag{2}$$

where X represents the input raw audio, h_i the output of the i–th transformer layer, and w_i learnable weights.

B. Attention-based fusion module (AFM)

In this work, we use AFM [21] to fuse two sets of features. AFM takes the concatenation of adjacent feature maps x and y as the input and yields the output, given by

$$F = (W(x, y) + 1) \cdot x + (1 - W(x, y)) \cdot y. \tag{3}$$

The local attention weight W in this context is computed by

$$W = \tanh\left(\text{BN}\left(W_2 \cdot \text{SiLU}\left(\text{BN}\left(W_1 \cdot [x, y]\right)\right)\right)\right), \tag{4}$$

where $[\cdot]$ denotes the concatenation along the channel dimension, W_1 and W_2 point-wise convolution with output channel sizes of C/r (r is the ratio for channel reduction) and C, respectively, and BN the batch normalization [22].

III. Proposed Method

Fig. 1 shows the overall framework of the proposed method, which can be divided into two cascaded modules, i.e., the front-end feature extractor and back-end classifier. The latter is totally the same as the baseline. The former includes two branches, where one processes the blocked deep feature of the PTM and the other employs convolution operation to process FBank features. The AFM is finally used to deeply merge FBank and PTM features.

A. Fusing Blocked Deep Features of PTM

We use the base structure of the PTMs in this work, which has 12 transformer layers. To fully utilize the output representations of transformer layers, we choose to divide the stacked layers into deep and shallow parts based on their position: the first half of layers is considered as the shallow part, and the rest belongs to the deep part. We only perform block partitioning operations at the shallow layers, and the representations of deep layers are processed altogether.

As shown in Fig 1(a), let $h^{(l)} \in \mathbb{R}^{T \times F}$ denote the representation of l-th shallow transformer layer, where T and F represent the numbers of frames and frequency bins, respectively. We split it into N sub-blocks along the channel dimension as

$$h^{(l)} = \left\{h_0^{(l)}, h_1^{(l)}, \ldots, h_{N-1}^{(l)}\right\}, \quad h_n^{(l)} \in \mathbb{R}^{T \times (F/N)}, \tag{5}$$

where N decreases from bottom to top layers, i.e., bottom layers have more sub-blocks. For example, in experiments the best choice of N for six shallow layers is [6, 6, 3, 3, 2, 2].

We first calibrate each sub-block with a learnable weight $W_n^{(l)}$, and then concatenate the weighted features along the channel dimension to restore the original tensor size as

$$\tilde{h}^{(l)} = \left\{w_0^{(l)} \cdot h_0^{(l)}, w_1^{(l)} \cdot h_1^{(l)}, \ldots, w_{N-1}^{(l)} \cdot h_{N-1}^{(l)}\right\}. \tag{6}$$

TABLE I
EXPERIMENTAL RESULTS OF DIFFERENT MODELS GIVEN 3S-DURATION TRAILS ON THE VOXCELEB1 EVALUATION SETS.

Front-end Model	Type	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H	
		EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF
Wav2Vec2.0 Base	Baseline	1.505	0.182	1.513	0.166	2.884	0.265
	Proposed	1.291	0.162	1.332	0.150	2.587	0.250
HuBERT Base	Baseline	1.409	0.181	1.496	0.160	2.921	0.281
	Proposed	1.261	0.177	1.327	0.145	2.524	0.248
WavLM Base+	Baseline	1.239	0.146	1.253	0.135	2.428	0.247
	Proposed	**0.984**	**0.117**	**1.140**	**0.129**	**2.264**	**0.234**

The calibrated representations of the shallow part are aggregated by linear regression. Since the representations of deep part have broader context information and capture global semantics, we skip channel splitting and calibration operations. Instead, we directly fuse representations of deep part via a linear regression operation. Finally, the features from the deep and shallow parts are aggregated using AFM as [21]

$$F_P = \text{Linear}\left\{\text{AFM}\left(\sum_{l=1}^{\frac{n}{2}} w_l \cdot \tilde{h}^{(l)}, \sum_{l=\frac{n}{2}+1}^{n} w_l \cdot h^{(l)}\right)\right\}. \quad (7)$$

For the base structure of the PTM with n = 12 transformer layers, $h^{(l)}$ denotes the representation of the l-th layer, and w_l the learnable weight.

B. Fusion with FBank Features

As shown in [19], [23], there exists a complementary relation between the hand-crafted and PTM features. In Fig 1(c), to leverage this complementarity, we extract Fbank features from the input audio and transform them into high-level features F_H using a lightweight CNN extractor. We then apply AFM to fuse these two sets of features, which are fed into the back-end ECAPA-TDNN to extract the x-vector embedding as

$$E = \text{ECAPA-TDNN}\left\{\text{Linear}\left(\text{AFM}(F_P, F_H)\right)\right\}. \quad (8)$$

IV. EXPERIMENTAL SETUP

Datasets and evaluation metrics: To validate the efficacy of proposed method, we set up three short-duration scenarios using VoxCeleb1-O, VoxCeleb1-E and VoxCeleb1-H trials [24]. The difference between these trials and regular trials lies in that test utterances are truncated to 3-second segments with random starting points. We utilize the development set of VoxCeleb2 [25] which comprises 5994 speakers for training. For data augmentation, we use MUSAN[26], room impulse response (RIR)[27] and speed perturbation[28] with 0.9 and 1.1 times speed changes to treble the number of speakers.

The performance is evaluated in terms of EER and minimum of the normalized detection cost function (MinDCF) with P_{target} = 0.01 and C_{fa} = C_{miss} = 1.

Configuration: We consider three SOTA PTMs: Wav2vec2.0, Hubert, and WavLM, all of which take raw audios as input and consist of seven convolutional layers and multiple transformer layers. During training the parameters of PTM are fixed, and we set the batch size to 128 and the learning rate to 1e-3. During the unfixed stage, we set the batch size to 64 and the learning rate to 5e-5. Adam optimizer is used with a decay of 1e-4 and the margin and scaling factors of AAM-Softmax loss [29] are set to 0.2 and 32, respectively. Experiments employ model averaging over the models from last three epochs. The final speaker embedding is extracted with a dimension of 256. We also use cosine similarity and adaptive s-norm [30], [31] to normalize the scores, where imposter cohort size is set to 300.

V. RESULTS AND ANALYSIS

A. Comparison with baseline

The main experimental results are shown in the Table I. It is clear that all the PTMs distinctly benefit from the proposed method. Compared to the baseline, the proposed method achieves a relative EER reduction of 14.2%, 11.5%, 20.6% by using Wav2Vec2.0, Hubert, WavLM as front-end on the short duration Vox1-O dataset,respectively. Among all the results, the proposed method using WavLM Base + as front-end achieves the best performance, resulting in EERs of 0.984%, 1.140%, and 2.264% on the three test sets, respectively. Thus we use this model in the following experiments and call it **Proposed**.

B. Comparison with different partition configurations

To obtain more discriminative features, we apply different partition operations to the shallow and deep parts of the PTM. In this section, we analyze the influence of various partition configurations on the overall performance.

We first investigate partition configurations to the six Transformer layers in the shallow part. From the results listed in table II, we can see the best performance is obtained with configuration of [6 6 3 3 2 2]. This means we should partition the bottom layers into more sub-blocks than the top layers. Nevertheless, an excessive number of overly fine-grained sub-blocks does not translate into enhanced results. In the following experiment, we investigate applying partition operation to the deep part. The experimental results are listed in Table III.

TABLE II
IMPACT OF SUB-BLOCKS FOR SHALLOW LAYERS WITH THE NUMBER FOR DEEP LAYERS SET TO BE [1 1 1 1 1 1].

Shallow Sub-block Nums	EER(%) 3s		
	Vox1-O	Vox1-E	Vox1-H
[1 1 1 1 1 1]	1.154	1.285	2.315
[6 6 6 6 6 6]	1.048	1.161	2.278
[2 2 3 3 6 6]	1.085	1.186	2.339
[6 6 3 3 2 2]	**0.984**	**1.140**	**2.264**
[12 6 3 3 2 2]	1.093	1.155	2.292

TABLE III
IMPACT OF SUB-BLOCKS IN DEEP LAYERS WITH THE NUMBER FOR SHALLOW LAYERS SET TO BE [6 6 3 3 2 2].

Deep Sub-block Nums	EER(%) 3s		
	Vox1-O	Vox1-E	Vox1-H
[6 6 3 3 2 2]	1.037	1.152	2.316
[2 2 2 2 2 2]	1.048	1.181	2.351
[1 1 1 1 1 1]	**0.984**	**1.140**	**2.264**

We can find that any partition to the deep layers results in performance degradation, so we don't apply any partition to the deep layers. The experimental results in above two Tables consistently reveal that more sub-blocks in shallow layers and less sub-blocks in deep layers can improve the SV performance.

C. Ablation experiments

Three sets of features, the shallow parts of PTM, the deep parts of PTM and FBank features are used in the proposed method. In this section, we design ablation studies to analyze the impact of different features on the final performance. The results in table IV indicates that the removal of any feature results in a noticeable performance decrease. The experimental results demonstrate that both the handcrafted features and the universal representations from different layers of the PTMs are complementary to SV task.

D. Comparison with SOTA Approaches

Finally, we compare the proposed method with some published systems in Table V. It is evident that our method achieves the best results in VoxCeleb datasets on short-duration. To our knowledge, WavLM Base+ FFPTM+DBE[19] is the SOTA approach. Compared to second best system, the proposed method can relatively decrease the EER by 11.0%, 4.3%, 5.2% on the three test sets, respectively.

Nevertheless, we also include the performance with normal duration (standard trials). The results in Table VI show the superiority of the proposed method. When combined with the proposed WavLM Base+ model, it can even approximate some large structured models on the Vox1-O dataset.

VI. CONCLUSION

In this work, we proposed a fusion method to jointly use the multi-layer features from PTM and hand-crafted features for SV. We partitioned shallow Transformer layers of PTM into sub-blocks and calibrate them to obtain more-refined deep features. The aggregated feature from PTM and hand-crafted features are fed into the backend classifier. Experimental results confirmed the effectiveness of the proposed method in standard-duration and short-duration SV tasks. In the future, we will further explore the application of blocked deep features from PTM to more speech-related tasks.

TABLE IV
IMPACT OF DIFFERENT FEATURES FED INTO THE BACK-END, WHERE "W/O" DENOTES "WITHOUT".

Architecture	EER(%) 3s		
	Vox1-O	Vox1-E	Vox1-H
Proposed	**0.984**	**1.140**	**2.264**
w/o FBank	1.090	1.181	2.385
w/o Shallow Features	1.175	1.294	2.480
w/o Deep Features	1.101	1.228	2.414

TABLE V
COMPARISON WITH OTHER SOTA MODELS ON VOXCELEB 1, WHERE TDNN IS SHORT FOR ECAPA-TDNN.

Architecture	EER(%) 3s		
	Vox1-O	Vox1-E	Vox1-H
WavLM Base+ - TDNN[16]	1.239	1.253	2.428
HuBERT Base - TDNN[16]	1.409	1.496	2.921
WavLM Base+-MFHA[32]	1.138	1.231	2.415
WavLM Base+ FFPTM+DBE [19]	1.106	1.191	2.387
WavLM Base+ (Proposed)	**0.984**	**1.140**	**2.264**

TABLE VI
COMPARISON WITH OTHER MODELS ON VOXCELEB 1.

Architecture	EER(%) normal duration		
	Vox1-O	Vox1-E	Vox1-H
HuBERT Base-MHFA [33]	0.88	1.06	2.11
WavLM Base+ FFPTM-TDNN[19]	0.776	0.853	1.775
WavLM Base+-MFHA[32]	0.660	0.890	1.900
Uni-SAT Large-TDNN[18]	0.696	**0.685**	**1.433**
WavLM Base+ (Proposed)	**0.643**	0.834	1.712

REFERENCES

[1] N. Singh *et al.*, "Voice biometric: A technology for voice based authentication," *Advanced Science, Engineering and Medicine*, vol. 10, no. 7-8, pp. 754–759, 2018.

[2] I. McGraw, R. Prabhavalkar, R. Alvarez, *et al.*, "Personalized speech recognition on mobile devices," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5955–5959. DOI: 10.1109/ICASSP.2016.7472820.

[3] J. P. Campbell, W. Shen, W. M. Campbell, R. Schwartz, J.-F. Bonastre, and D. Matrouf, "Forensic speaker recognition," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 95–103, 2009. DOI: 10.1109/MSP.2008.931100.

[4] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056. DOI: 10.1109/ICASSP.2014.6854363.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333. DOI: 10.1109/ICASSP.2018.8461375.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[7] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proceedings of Interspeech*, 2020, pp. 3830–3834.

[8] T. Liu, R. K. Das, K. Aik Lee, and H. Li, "Mfa: Tdnn with multi-scale frequency-channel attention for text-independent speaker verification with short utterances," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7517–7521. DOI: 10.1109/ICASSP43922.2022.9747021.

[9] Y. Chen, S. Zheng, H. Wang, L. Cheng, Q. Chen, and J. Qi, "An enhanced res2net with local and global feature fusion for speaker verification," in *Proceedings of Interspeech*, 2023, pp. 2228–2232.

[10] H.-J. Heo, U.-H. Shin, R. Lee, Y. Cheon, and H.-M. Park, "Next-tdnn: Modernizing multi-scale temporal convolution backbone for speaker verification," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 11 186–11 190. DOI: 10.1109/ICASSP48485.2024.10447037.

[11] Y. Sun, C. Li, and B. Li, "Branchformer-based tdnn for automatic speaker verification," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 10 981–10 985. DOI: 10.1109/ICASSP48485.2024.10448107.

[12] R. K. Das and S. R. M. Prasanna, "Speaker verification for variable duration segments and the effect of session variability," in *Advances in communication and computing*, 2015, pp. 193–200.

[13] R. K. Das and S. R. M. Prasanna, "Speaker verification from short utterance perspective: A review," *IETE Technical Review*, vol. 35, no. 6, pp. 599–617, 2017. DOI: 10.1080/02564602.2017.1357507.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

[15] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. 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. DOI: 10.1109/TASLP.2021.3122291.

[16] S. Chen, C. Wang, Z. Chen, *et al.*, "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. DOI: 10.1109/JSTSP.2022.3188113.

[17] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[18] Z. Chen, S. Chen, Y. Wu, *et al.*, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6147–6151. DOI: 10.1109/ICASSP43922.2022.9747814.

[19] S. Peng, W. Guo, H. Wu, Z. Li, and J. Zhang, "Fine-tune pretrained models with multi-level feature fusion for speaker verification," in *Proceedings of Interspeech 2024*, 2024, pp. 2110–2114. DOI: 10.21437/Interspeech.2024-260.

[20] G. Gábor, T. László, S. Veronika, *et al.*, "Investigating the utility of wav2vec 2.0 hidden layers for detecting multiple sclerosis," in *Proceedings of SPECOM 2024*, Belgrade, Serbia, Nov. 2024, pp. 297–308.

[21] J. Qi, W. Guo, and B. Gu, "Bidirectional multiscale feature aggregation for speaker verification," in *Proceedings of Interspeech*, 2021, pp. 71–75.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, ser. JMLR Workshop and Conference Proceedings, vol. 37, JMLR.org, 2015, pp. 448–456.

[23] Y. Li *et al.*, "Efficient integrated features based on pre-trained models for speaker verification," in *Proc. Interspeech*, 2024.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[25] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[26] D. Snyder, G. Chen, and D. Povey, *Musan: A music, speech, and noise corpus*, CoRR, vol. abs/1510.08484, 2015.

[27] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224. DOI: 10.1109/ICASSP.2017.7953152.

[28] W. Wang, D. Cai, X. Qin, and M. Li, *The dku-dukeece systems for voxceleb speaker recognition challenge 2020*, arXiv preprint arXiv:2010.12731, 2020.

[29] 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 (CVPR)*, Jun. 2019.

[30] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4512–4515. DOI: 10.1109/ICASSP.2011.5947357.

[31] S. Cumani, P. D. Batzu, D. Colibro, C. Vair, P. Laface, V. Vasilakakis, *et al.*, "Comparison of speaker recognition approaches for real applications," in *Proc. Interspeech*, 2011, pp. 2365–2368.

[32] J. Peng, O. Plchot, T. Stafylakis, L. Mošner, L. Burget, and J. Cernocky, "An attention-based backend allowing efficient finetuning of transformer models for speaker verification," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2023, pp. 555–562.

[33] J. Peng, O. Plchot, T. Stafylakis, L. Mošner, L. Burget, and J. Cernocky, "Improving speaker verification with self-pretrained transformer models," in *Proc. Interspeech*, 2023, pp. 5361–5365.