CA-MHFA: A Context-Aware Multi-Head Factorized Attentive Pooling for SSL-Based Speaker Verification

Junyi Peng[1], Ladislav Mošner[1], Lin Zhang[1], Oldřich Plchot[1], Themos Stafylakis[2,3], Lukáš Burget[1], Jan Černocký[1]
[1]Speech@FIT, Brno University of Technology, Czechia
[2]Athens University of Economics and Business, Greece
[3]Omilia, Archimedes/Athena R.C., Greece

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10889058

***Abstract*—Self-supervised learning (SSL) models for speaker verification (SV) have gained significant attention in recent years. However, existing SSL-based SV systems often struggle to capture local temporal dependencies and generalize across different tasks. In this paper, we propose context-aware multi-head factorized attentive pooling (CA-MHFA), a lightweight framework that incorporates contextual information from surrounding frames. CA-MHFA leverages grouped, learnable queries to effectively model contextual dependencies while maintaining efficiency by sharing keys and values across groups. Experimental results on the VoxCeleb dataset show that CA-MHFA achieves EERs of 0.42%, 0.48%, and 0.96% on Vox1-O, Vox1-E, and Vox1-H, respectively, outperforming complex models like WavLM-TDNN with fewer parameters and faster convergence. Additionally, CA-MHFA demonstrates strong generalization across multiple SSL models and tasks, including emotion recognition and anti-spoofing, highlighting its robustness and versatility.** [1]

***Index Terms*—Self-supervised learning, speaker verification, speaker extractor, pooling mechanism, speech classification**

I. Introduction

Large-scale self-supervised learning (SSL) speech models, such as Wav2vec [1], HuBERT [2], WavLM [3], and their variants [4], have significantly advanced various speech-related tasks, including speech recognition (ASR) [5], emotion recognition (ER) [6], deepfake detection [7], and speaker verification (SV) [8], [9]. These SSL models are pre-trained on extensive speech datasets, which allow them to learn universal representations that are more robust to channel mismatch and noisy conditions compared to traditional acoustic features like Mel-frequency cepstral coefficients (MFCCs) and FBank, especially under low-resource scenarios [10], [11].

The most common approach to adapting these general-purpose models to specific downstream tasks is by fine-tuning the entire pre-trained SSL model with a task-oriented back-end module using labeled downstream data. In the field of SV, since the outputs of SSL models are layer-wise frame-by-frame representations, the task-oriented module (known as the speaker extractor back-end) processes these features to generate utterance-level speaker embeddings, which are then used for speaker identity prediction. It typically includes a frame-level feature extractor, a pooling mechanism, and an utterance-level feature extractor. For example, in [8], [12], a weighted sum of layer-wise SSL outputs replaces traditional acoustic features as input to speaker extractors, such as the x-vector [13] and the ECAPA-TDNN models [14]. These SSL-based SV systems significantly accelerate convergence and boost performance compared to those based on MFCC features. To further explore the potential of SSL models, a lighter back-end, called multi-head factorized attentive pooling (MHFA) [15], was proposed. MHFA employs two sets of weights to model different aspects of input features. This approach has been shown to outperform systems that integrate ECAPA-TDNN on the VoxCeleb dataset.

Although MHFA has achieved promising results on the SV task, it typically operates at the utterance level. By processing all frames simultaneously, it ignores the detailed relationships between surrounding frames. This lack of local context limits its ability to capture dynamic and fine-grained temporal dependencies across frames. Additionally, since pre-trained SSL models already possess strong frame-level modeling capabilities by leveraging self-attention mechanisms, incorporating complex and randomly initialized frame-level modules in the back-end can mislead the optimization process, leading to suboptimal performance [16]. Furthermore, studies on the generalizability of these lightweight back-end modules to other speech classification tasks, such as emotion recognition (ER) and spoofing detection, remain limited. This raises the concern that these modules might be over-optimized for a single task, such as SV [17].

To address the aforementioned challenges, in this paper, we aim to develop a lightweight framework considering context information for SV and can be applied to broader speech classification tasks and various SSL models. The proposed framework, named context-aware multi-head factorized attentive pooling (CA-MHFA), utilizes both past and future speech frames as contextual information to extract utterance-level representations from SSL layer-wise features. Specifically, similar to the self-attention mechanism used in MHFA [15], we employ two sets of learnable weights along with a linear layer to generate *keys* and *values*. Unlike MHFA, we introduce a set of learnable *queries* that are grouped, allowing the model to focus on the surrounding frames and better capture contextual dependencies. The attention weights are computed using convolution between the grouped *queries* and *keys*. Finally, an attentive pooling layer aggregates all frames from each group, followed by a concatenation and linear layer to compute the speaker embedding. Moreover, we share the single key and value with the grouped queries to simplify the entire pipeline.

The contributions of our work are as follows:

- **A lightweight back-end module:** We propose a lightweight back-end, CA-MHFA, to extract representations that consider contextual information from nearby frames when calculating attention weights.
- **Compatible architecture:** Besides MHFA, our context-aware extension is compatible with other pooling methods, like mean pooling and attentive pooling [18], making it extensible for future integration and improvements to existing ones.
- **State-of-the-art SV performance:** We achieve SOTA results on the VoxCeleb dataset using fewer model parameters. With the same pre-trained SSL model, the proposed system outperforms the WavLM-TDNN [3] and yields 0.42%, 0.48% and 0.96% EER on Vox1-O, Vox1-E and Vox1-H, respectively.
- **Strong generalization:** We demonstrate the effectiveness of the proposed lightweight back-end module across various speech classification tasks and SSL models, including SV, ER, and

[1] Code is available at https://github.com/BUTSpeechFIT/wespeaker_ssl_public

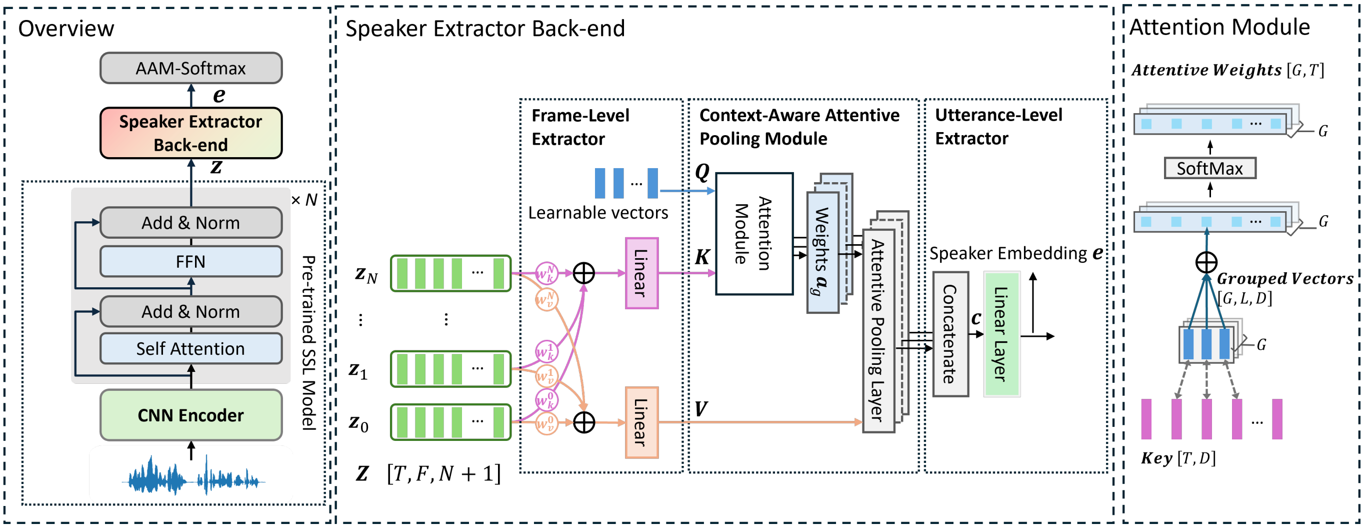

Fig. 1. The architecture of the pre-trained model and attached lightweight speaker extractor back-end (context-aware multi-head factorized attentive pooling, CA-MHFA). During the fine-tuning, the SSL model and cascaded speaker extractor are jointly optimized.

deepfake detection, following the SUPERB principle. This is evaluated by integrating with nine popular pre-trained SSL models, including Wav2Vec 2.0, HuBERT, Data2vec, and WavLM.

II. Context-aware Multi-head Factorized Attentive Pooling

In this section, we describe the proposed CA-MHFA in detail. As shown in Fig. 1, it consists of three main components: a frame-level extractor, a context-aware attentive pooling module, and an utterance-level extractor.

A. Frame-Level Extractor with Compression

The goal of text-independent speaker embedding extractors is to capture speaker-related characteristics independent of content. However, SSL models are designed as general-purpose models, not specifically for speaker embedding extraction. As a result, the representations they extract often contain a mix of both speaker-related and content-related information [12]. Although the weighted sum method [12], which uses learnable weights for each layer, can help mitigate this issue, it treats content and speaker information within the same layer equally. Consequently, when the method reduces the weights for layers rich in content-related information, it also unintentionally reduces the speaker-related information from those layers. To effectively utilize and refine speaker-related information from all layers, the proposed CA-MHFA is designed to factorize SSL representations into multiple subsets, focusing on extracting speaker-specific features while minimizing interference from content.

To describe the proposed CA-MHFA, we use the *query/key/value* abstraction for the attention mechanism [19]. For a SSL model, we define outputs from its different layers as $\mathbf{Z} = \{\mathbf{z}_0, ..., \mathbf{z}_N\}$, $\mathbf{z}_n \in \mathbb{R}^{T \times F}$, where T is the dimention of the frame level, F is the feature dimension of outputs from Transformer blocks, and N denotes the number of Transformer blocks inside the SSL model. To construct multiple subsets, we employ two sets of normalized weights (factors) ω_n^k and ω_n^v along with a linear layer to separately aggregate and compress layer-wise outputs to produce the matrices of *keys* $\mathbf{K}$ and *values* $\mathbf{V}$, respectively:

$$\begin{aligned}\mathbf{K} &= \left(\sum_{n=0}^{N} \omega_n^k \mathbf{z}_n\right) \mathbf{S}^k \\ \mathbf{V} &= \left(\sum_{n=0}^{N} \omega_n^v \mathbf{z}_n\right) \mathbf{S}^v,\end{aligned} \tag{1}$$

where $\mathbf{S}^k, \mathbf{S}^v \in \mathbb{R}^{F \times D}$ denote transformation matrices responsible for compressing the weighted frame-level features into keys and values, respectively, and D represents the feature dimension after compression.

B. Context-Aware Attentive Pooling

To effectively model the information within the key flow, we introduce a set of global, input-agnostic, learnable vectors, denoted as *queries* $\mathbf{Q} \in \mathbb{R}^{LG \times D}$. For contextual modeling, queries $\mathbf{Q}$ are divided into G groups, with each group's vectors denoted as $\mathbf{q}^g = [\mathbf{q}_1^g, .., \mathbf{q}_L^g]$, $\mathbf{q}^g \in \mathbb{R}^{L \times D}$, where $g \in [1, G]$ indexes the groups (heads) and each group $\mathbf{q}^g$ comprises L trainable vectors. For computational efficiency, each of the groups $\mathbf{q}^g$ shares the same key and value.

To model contextual information, we consider R frames on either side of the current frame, resulting in a total of L neighboring frames used to compute attention weights for the current frame. The current frame refers to the frame in focus during attention calculation, with neighboring frames providing additional context information. Here, $R = \text{Floor}((L-1)/2)$. The attention weights of each group $\mathbf{a}^g \in \mathbb{R}^{T \times 1}$ are further computed by $\mathbf{q}^g$ and $\mathbf{K}$ as follows:

$$a_t^g = \frac{\exp\left(\frac{1}{L}\sum_{j=-R}^{R} \mathbf{q}_j^g \mathbf{k}_{t+j}^\top\right)}{\sum_{i=1}^{T} \exp\left(\frac{1}{L}\sum_{m=-R}^{R} \mathbf{q}_m^g \mathbf{k}_{i+m}^\top\right)}, \tag{2}$$

where a_t^g represents the attentive weight given to the current t-th frame after considering the surrounding frames L, and $\top$ denotes transpose. From an implementation point of view, this process can be viewed as a 2D convolution operation. Specifically, the input $\mathbf{K}$ consists of D channels, and the output channels correspond to the number of groups G. The kernel size is (L, D). In this way,

TABLE I

HYPER-PARAMETER ANALYSIS OF CA-MHFA WITH DIFFERENT CONTEXT LENGTHS L AND NUMBERS OF HEADS G IN TERMS OF EER (%) ON VOXCELEB DATASET. † DENOTES THE SYSTEM APPLYING CONVOLUTION IN BOTH *keys* AND *values*.

Methods	#Param	#Context	#Head	Vox1-O	Vox1-E	Vox1-H
MHFA	0.72M	1	16	0.79	0.85	1.71
MHFA	1.25M	1	32	0.79	0.81	1.61
MHFA	2.30M	1	64	0.76	0.79	1.58
CA-MHFA	1.25M	3	32	0.76	0.76	1.54
CA-MHFA	1.26M	5	32	0.73	0.76	1.53
CA-MHFA	2.30M	3	64	0.73	0.74	1.49
CA-MHFA	2.31M	5	64	**0.69**	0.74	1.47
CA-MHFA	2.36M	9	64	0.70	**0.72**	**1.45**
CA-MHFA	2.41M	17	64	0.69	0.74	1.47
CA-MHFA†	2.32M	5	64	0.72	0.76	1.52

TABLE II

PERFORMANCE COMPARISON WITH THE PROPOSED CA-MHFA WITH OTHER FULL FINE-TUNED SSL-BASED SYSTEMS, AS WELL AS SPEAKER EXTRACTORS TRAINED FROM SCRATCH. **ET.** DENOTES THE ECAPA-TDNN MODEL. † DENOTES WESPEAKER'S IMPLEMENTATION. ‡ REPRESENTS OUR IMPLEMENTATION.

Methods	FLOPs	#Param	Vox1-O	Vox1-E	Vox1-H
			EER	EER	EER
ECAPA-TDNN (ET.) [20]	1.04G	6.19M	0.90	1.11	2.32
ResNet221 [21]	21.29G	23.79M	0.50	0.67	1.21
ResNet293 [21]	28.10G	28.62M	0.44	0.65	1.18
WavLM_BASE_Plus + ET. [3]	-	94M + 6M	0.84	0.92	1.75
NEMO Large + MFA [22]	-	130M	0.48	0.71	1.54
Conformer + MHFA [10]	-	172M + 2.3 M	0.65	0.93	1.86
UniSpeech-SAT_Large + ET. [8]	-	316M + 6M	0.53	0.56	1.18
WavLM_Large + ET. [3]	-	316M + 6M	0.38	0.48	0.98
WavLM_Large + ET. [3]†	-	316M + 6M	0.41	0.55	1.11
WavLM_Large + MHFA [15]‡	25.79G	316M + 2.3M	0.55	0.59	1.24
WavLM_Base_Plus + CA-MHFA	11.05G	94M + 2.3M	0.70	0.72	1.45
+ LMF/QMF	11.05G	94M + 2.3M	0.59	0.65	1.30
WavLM_Large + CA-MHFA	25.79G	316M + 2.3M	0.55	0.62	1.18
+ LMF/QMF	25.79G	316M + 2.3M	0.42	0.48	0.96

$\mathbf{Q}$ is reshaped into grouped vectors and serves further as the G convolutional kernels, each with the shape (L, D).

C. Utterance-Level Extractor

Next, each group's speaker representations $\mathbf{c}^g \in \mathbb{R}^{1\times D}$ are aggregated via attention weights $\mathbf{a}^g$ and then concatenated as:

$$\begin{aligned} \mathbf{c}^g &= \sum\nolimits_{t=1}^{T} \mathbf{a}_t^g \mathbf{v}_t, \\ \mathbf{c} &= \text{concat}\left(\mathbf{c}^1, ..., \mathbf{c}^G\right), \end{aligned} \tag{3}$$

where $\mathbf{c} \in \mathbb{R}^{1\times GD}$ aggregates the speaker representation at the utterance level from all subspaces (heads). Finally, the prediction of the speaker label is performed by passing the representation through a subsequent linear layer, L_2 normalization, and a classification layer, which is needed only during training. During testing, the output of the L_2 normalized linear layer is considered as the *speaker embedding* $\mathbf{e}$.

D. Generalization of Other Works

The CA-MHFA extends the MHFA framework by introducing flexibility in the attention mechanism. When we set $L = 1$, CA-MHFA simplifies to the original MHFA as presented in [15].

In scenarios utilizing average pooling, setting $\mathbf{q}_l^g$ to a constant non-trainable zero vector implies that the similarity of each frame to every group query equals zero. After softmax normalization, this results in equal weights of $1/T$ for each frame feature, in this case, CA-MHFA degenerates to average mean pooling.

Compared to self-attentive pooling [18], setting $G = 1$ implies that $\mathbf{Q}$ comprises a single group that contains only one D-dimensional trainable query. In this configuration, only a single set of weights is computed, making CA-MHFA equivalent to self-attentive pooling.

III. EXPERIMENTS

A. Setup

1) Two Types of Evaluation: We explored two types of evaluation: 1) Full fine-tuning condition: Both the SSL and back-end modules are trainable during fine-tuning. 2) SUPERB-style: We follow the SUPERB principle [12] and only update the back-end module during the fine-tuning stage, while keeping the SSL model frozen.

2) Datasets: We first focus on exploring the efficiency of the lightweight CA-MHFA for the SV task, involving two of the aforementioned evaluation methods: 1) For full fine-tuning conditions, to ensure we have enough data for finetuning SSL model, we use the development set of VoxCeleb2 for training, then use *Vox1-O*, *Vox1-E*, and *Vox1-H* trials for evaluating. 2) For SUPERB-style, the training set is VoxCeleb1 and the evaluation set is *Vox1-O* following [12]. We also explored the generalizability of CA-MHFA to two other speech classification tasks based on the SUPERB-style evaluation: For ER, the models are trained and evaluated on the IEMOCAP corpus [23], which consists of approximately 12 hours of recordings in five sessions. Regarding deepfake detection, we use the ASVspoof 2019 LA [24] dataset.

3) Implementation details: For all three aforementioned tasks, we utilize two scales of pre-trained SSL models listed in Table III: 1) The *BASE* models, consisting of a CNN encoder and 12 layers of Transformers with approximately 94M parameters. The dimension of the Transformer output F is 768. 2) The *Large* models, which include 24 layers of Transformers with approximately 316M parameters. In the full fine-tuning configurations, the extracted speaker embedding dimension is 256. For the SV task under full fine-tune condition, we employ the AAM-softmax [25] loss function with a margin of 0.2 and a scaling factor of 32. During gradient updates, the gradients of the SSL model are scaled by 0.1. To further boost performance, we adopt large margin tuning [26] with longer (5-second) segments and a margin of 0.5 for an additional 3 training epochs. The initial learning rate is set to 1e-4 and decreases to 1e-6 by the 10th epoch using the AdamW optimizer. All fine-tuning datasets are augmented with with MUSAN [27] and room impulse responses. In addition, we used speaker-wise adaptive score normalization and Quality Measure Functions [26] to calibrate the scores.

When evaluating the generalizability of the model across different tasks and upstream models, we adopt the SUPERB-style evaluation: For SV, all systems use the AM-softmax loss function. To keep consistent with the x-vector implementation within the SUPERB, the speaker embedding dimensions of ECAPA-TDNN, MHFA, and CA-MHFA, are set to 512. The MHFA model uses 8 heads, while the CA-MHFA model uses 8 heads with a context length of 9. For ER, cross-entropy (CE) loss is employed for optimization, and mean pooling is performed via a weighted sum across layer-wise features, followed by mean pooling along the time dimension and a final linear layer. For deepfake detection, CE loss is used, with a learning rate set to 1e-4 using the Adam optimizer, following [7]. For the back-end module, we use the weighted sum with LLGF (LCNN-BLSTM) when MHFA is not utilized, as this structure has shown the best performance when the SSL model is frozen [7]. No augmentation methods are applied.

TABLE III
COMPARISON OF DIFFERENT FROZEN UPSTREAM MODELS ACROSS THREE DOWNSTREAM TASKS: SV, ER, AND DEEPFAKE DETECTION WITH VARYING BACK-END MODELS FOLLOWING SUPERB PRINCIPLE.

Upstream	SV EER(%)↓				Emotion Recognition ACC(%)↑			Deepfake Detection EER(%)↓		
	x-vector	ECAPA-TDNN	MHFA	CA-MHFA	MeanPooling [12]	MHFA	CA-MHFA	LLGF	MHFA	CA-MHFA
Wav2vec2.0 BASE	5.43	3.56	3.11	3.07	60.55	62.90	63.13	2.03	0.73	0.52
HuBERT BASE	5.65	3.29	2.78	2.82	60.73	62.35	63.96	2.39	1.19	1.22
Data2vec BASE	6.39	3.97	3.38	3.26	65.43	65.93	65.52	3.83	1.53	1.37
WavLM BASE	4.84	3.04	2.45	2.41	62.48	65.06	67.64	2.13	0.93	0.73
WavLM BASE Plus	4.10	2.67	1.87	1.79	66.72	66.58	66.45	3.64	0.20	0.30
Wav2vec2.0 Large	6.06	2.86	2.72	2.64	62.76	63.59	64.42	0.83	0.68	2.39
HuBERT Large	6.02	2.71	2.43	2.34	62.48	63.72	64.97	2.30	0.75	0.67
Data2vec Large	7.62	3.11	2.59	2.62	64.97	64.88	64.79	4.26	1.41	1.32
WavLM Large	4.87	2.17	1.78	1.77	67.92	69.72	71.52	1.54	2.23	1.21

4) Performance Metrics: For SV and deepfake detection, equal error rate (EER) is employed to measure the performances. For ER, we use a leave-one-session-out 5-fold cross-validation to report averaged accuracy.

B. Hyper-Parameter Analysis of CA-MHFA

We present here a hyper-parameter analysis of CA-MHFA for the SV task under the full fine-tuning evaluation. We focus on analyzing different context lengths (L) and numbers of heads (G) in terms of EER (%) on the VoxCeleb dataset. WavLM BASE Plus is used as the pre-trained SSL model [3]. The results are shown in Table I.

It is observed that increasing the context length and the number of heads improves the performance of CA-MHFA, especially on Vox1-E and Vox1-H. This may be due to the fact that while the SV task does not require extensive context, considering adjacent surrounding frames can be beneficial. Compared to MHFA, CA-MHFA consistently outperforms MHFA in most configurations.

We also compared systems that applied convolution operations to both the *keys* and *values* branches, as shown in the last row of Table I. Specifically, the *values* branches used 1D convolution operations with a kernel size of 5, where the input and output channels were set to D. The results show that this configuration performs worse compared to systems that applied convolution only to the *keys* branch. This suggests that contextual information may be more beneficial for the content-related subspace than for the speaker-related subspace. One possible explanation is that context in the content subspace helps capture the speaker's speaking style, such as the structuring of words, while the speaker subspace focuses on features that are invariant to the speaker's identity and, thus, requires less contextual attention.

C. Comparison with State-of-the-art SV Systems

Under the optimal hyperparameters identified in the previous section ($G = 64$, $L = 9$), we compared the proposed CA-MHFA with other state-of-the-art SV systems, including both SSL-based models and speaker extractors trained from scratch, as shown in Table II.

It is observed that WavLM_Large combined with CA-MHFA outperforms the SOTA model trained from scratch like ECAPA-TDNN and deep ResNet models (e.g., ResNet221 and ResNet293) with fewer training steps (23 epochs vs. 150 epochs). Additionally, despite the WavLM_Large model having a much larger number of parameters than ResNet293, the FLOPs are comparable when using 2 seconds of speech as input. This indicates that after convergence, the inference speed of WavLM_Large + CA-MHFA is comparable to that of ResNet293. Furthermore, when comparing SSL models of similar scales, WavLM + CA-MHFA consistently outperforms other SSL-based SV systems with a lighter speaker extractor back-end. For example, under the same WeSpeaker framework, the proposed CA-MHFA outperforms ECAPA-TDNN systems when using both WavLM_Large and WavLM_Base_Plus.

D. CA-MHFA on Other Tasks

In this section, we explored the generalizability of the proposed CA-MHFA across three downstream tasks: SV, ER, and deepfake detection. The evaluation follows the SUPERB-style as we introduced in section III-A. The results are summarized in Table III.

In general, the proposed CA-MHFA consistently improves performance in multiple tasks and upstream models over baseline systems and MHFA. Specifically, for SV, when using the same SSL features as input, CA-MHFA consistently outperforms the x-vector and ECAPA-TDNN back-ends with fewer parameters (2.3M vs. 9.2M and 7.0M, respectively). In addition, CA-MHFA shows notable performance improvements in ER tasks. In the deepfake detection, compared to LLGF, CA-MHFA demonstrates Considerable improvements in most cases. Our results demonstrate consistent improvements across different model scales, highlighting its robustness and versatility.

IV. CONCLUSION

In this paper, we propose a novel lightweight back-end module named CA-MHFA for pre-trained SSL-based speaker verification, which can be easily extended to broader speech classification tasks, including emotion recognition and deepfake detection. CA-MHFA enhances contextual modeling by considering frames surrounding the current frame when computing attention weights, leading to more stable and discriminative utterance-level representations. We conducted comprehensive experiments on the VoxCeleb, IEMOCAP, and ASVspoof2019 datasets. The results demonstrate that CA-MHFA outperforms other downstream back-end modules while maintaining lower parameter counts.

V. ACKNOWLEDGMENT

The work was supported by European Defense Fund's project No. 101168083 "ARCHER", Czech Ministry of Interior project No. VJ01010108 "ROZKAZ", Ministry of Education, Youth and Sports of the Czech Republic (MoE) through the OP JAK project "Linguistics, Artificial Intelligence and Language and Speech Technologies: from Research to Applications"(ID: CZ.02.01.01/00/23_020/0008518) and Horizon 2020 Marie Sklodowska-Curie grant ESPERANTO, No. 101007666. Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports through the e-INFRA CZ (ID: 90254).

References

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[2] 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.

[3] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[4] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[5] Z. Li, T. Graave, J. Liu, T. Lohrenz, S. Kunzmann, and T. Fingscheidt, "Parameter-efficient cross-language transfer learning for a language-modular audiovisual speech recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[6] E. Morais, R. Hoory, W. Zhu, I. Gat, M. Damasceno, and H. Aronowitz, "Speech emotion recognition using self-supervised features," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6922–6926.

[7] X. Wang and J. Yamagishi, "Investigating Self-Supervised Front Ends for Speech Spoofing Countermeasures," in *Proc. Odyssey*, 2022, pp. 100–106.

[8] Z. Chen, S. Chen, Y. Wu, Y. Qian, C. Wang, S. Liu, Y. Qian, and M. Zeng, "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)*. IEEE, 2022, pp. 6147–6151.

[9] J. Peng, T. Stafylakis, R. Gu, O. Plchot, L. Mošner, L. Burget, and J. Černockỳ, "Parameter-efficient transfer learning of pre-trained transformer models for speaker verification using adapters," *arXiv preprint arXiv:2210.16032*, 2022.

[10] J. Peng, O. Plchot, T. Stafylakis, L. Mosner, L. Burget, and J. H. Černocký, "Improving Speaker Verification with Self-Pretrained Transformer Models," in *Proc. INTERSPEECH 2023*, 2023, pp. 5361–5365.

[11] P.-c. Hsu, A. Elkahky, W.-N. Hsu, Y. Adi, T. A. Nguyen, J. Copet, E. Dupoux, H.-y. Lee, and A. Mohamed, "Low-resource self-supervised learning with ssl-enhanced tts," *arXiv preprint arXiv:2309.17020*, 2023.

[12] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[13] 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)*. IEEE, 2018, pp. 5329–5333.

[14] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

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

[16] Z. Aldeneh, T. Higuchi, J.-w. Jung, S. Seto, T. Likhomanenko, S. Shum, A. H. Abdelaziz, S. Watanabe, and B.-J. Theobald, "Can you remove the downstream model for speaker recognition with self-supervised speech features?" *arXiv preprint arXiv:2402.00340*, 2024.

[17] S. Zaiem, Y. Kemiche, T. Parcollet, S. Essid, and M. Ravanelli, "Speech Self-Supervised Representation Benchmarking: Are We Doing it Right?" in *Proc. INTERSPEECH 2023*, 2023, pp. 2873–2877.

[18] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification," in *Proc. Interspeech 2018*, 2018, pp. 3573–3577.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Interspeech2020*, 2020, pp. 1–5.

[21] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[22] D. Cai and M. Li, "Leveraging asr pretrained conformers for speaker verification through transfer learning and knowledge distillation," *arXiv preprint arXiv:2309.03019*, 2023.

[23] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.

[24] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, "Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[26] J. Thienpondt, B. Desplanques, and K. Demuynck, "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5814–5818.

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