ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446845

IMPLICIT ENHANCEMENT OF TARGET SPEAKER IN SPEAKER-ADAPTIVE ASR THROUGH EFFICIENT JOINT OPTIMIZATION

Minghui Wu[1], *Haitao Tang*[2], *Jiahuan Fan*[2], *Ruoyu Wang*[1], *Hang Chen*[1], *Yanyong Zhang*[1], *Jun Du*[1,*], *Hengshun Zhou*[2], *Lei Sun*[2], *Xin Fang*[2], *Tian Gao*[2], *Genshun Wan*[2], *Jia Pan*[2], *Jianqing Gao*[2]

[1] University of Science and Technology of China, China [2] iFLYTEK Research, China

ABSTRACT

In multi-speaker scenarios, automatic speech recognition (ASR) models rely on pre-processed audio after speaker separation. However, when the target speaker is not accurately separated, ASR models face limitations in reaching their peak performance. To address this issue, we propose a speaker-adaptive ASR framework that possesses more implicit target speaker enhancement capability by efficiently joint-optimized speaker recognition (SR) and ASR models. Our framework introduces sharing self-supervised learning representation, optimization transfer and hierarchy speaker-gated attention. In this manner, it can maximize effectiveness of embedding bias and emphasize target speaker corresponding to semantic units. In the CHiME-7 DASR sub-track, the proposed method achieves a 28.19% relative reduction in word error rate (WER) on the development sets when compared to the official baseline. Notably, this framework has also been employed in the champion system for the CHiME-7 DASR.

Index Terms— CHiME-7 challenge, automatic speech recognition, speaker-adaptive, target speaker enhancement

1. INTRODUCTION

End-to-end (E2E) automatic speech recognition (ASR) models, especially encoder-decoder (ED) architectures [1, 2], have demonstrated impressive results in single-speaker scenarios. Yet, real-world situations frequently involve overlapping speech from multiple speakers, as seen in meetings and smart homes, posing challenges for traditional approaches [3]. The CHiME organizers have addressed this with a series of challenges [4]. In these competitions, researchers usually consider ASR and speech separation as distinct tasks [5]. Typically, they use guided source separation (GSS) methods in the front-end of ASR models to separate speakers [6]. However, GSS-based speech separation often includes residual non-target speaker components, which can impact the performance of ASR models.

To address these issues, the prevailing approach employs speaker adaptation techniques. These methods enhance ASR models' capability to implicitly adapt to the target speaker by integrating their embeddings into ASR input features [7, 8]. The method consists of two stages: Firstly, a pre-trained speaker recognition (SR) model based on ECAPA-TDNN is employed to extract offline speaker embeddings [9]. Secondly, the ASR input features are obtained by concatenating audio features with the extracted embeddings. The above stages present the following issues: 1) There is a mismatch in the acoustic scenario between pre-trained SR and ASR models [8]. 2) The statistics pooling (SP) layer in ECAPA-TDNN has a drawback of information loss, which leads to insufficient embedding bias for the second stage [10]. 3) The simple concatenation method only combines low-level features, disregarding the semantic representation of higher-level ones [11]. Consequently, this approach results in insufficient embedding bias strength, making it challenging to propagate into higher levels.

We introduce a framework called Speaker-Adaptive Implicit target Speaker enhancement (SAIS) to optimize both SR and ASR models efficiently. To reduce acoustic scene mismatch across different modules, we incorporate a self-supervised learning representation (SSLR) module known for its robustness [12, 13]. The SSLR module handles audio feature extraction and online speaker embeddings. We enhance ASR's ability to improve target speaker performance through a joint fine-tuning (JFT) process involving SSLR, SR, and ASR modules. Specifically, we replace the SP layer in ECAPA-TDNN with optimization transfer (OT) [14] to obtain more accurate bias information for the target speaker. OT minimizes information loss by constructing mapping and cost matrices for embeddings. Additionally, we introduce hierarchy speaker-gated attention (HSGA) to effectively integrate target speaker information at each encoder layer in the ASR module. These optimizations significantly boost ASR's implicit capability for enhancing target speaker performance.

2. SYSTEM DESCRIPTION

2.1. Speaker-adaptive ASR

To enhance the implicit target speaker capability of an ASR model, the speaker-adaptive ASR (SA-ASR) approach is commonly used. This involves concatenating audio features

*corresponding author

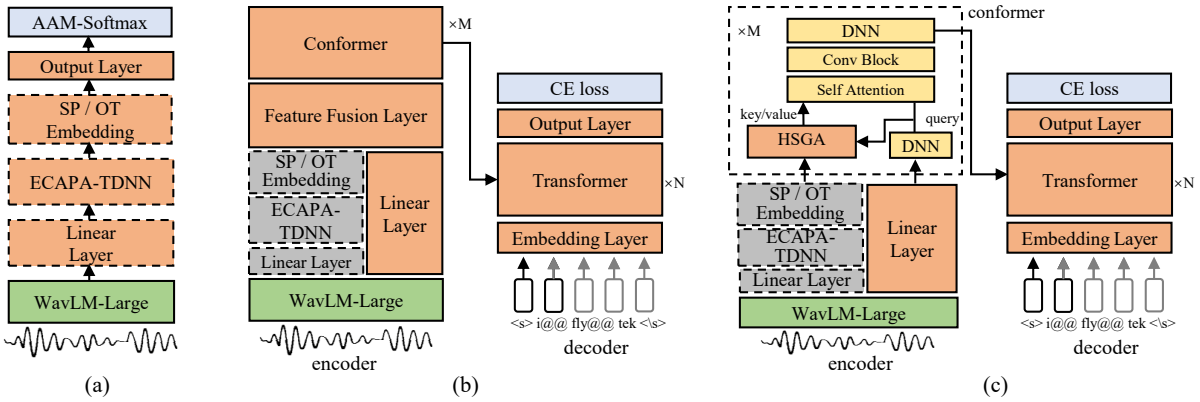

Fig. 1. (a) SSLR-based SR; (b) Sharing SSLR for SR and ASR; (c) SAIS architecture. In Embedding modules (a) to (c), there are two ways to generate x-vector for SR: traditional SP, proposed OT.

and speaker embeddings as input features for the ASR system. In our case, we employ an attention-based ED ASR module that utilizes a conformer network for encoding [15]. The decoder consists of an embedding layer, transformer network, and output layer [2]. For audio features, SSLR has been proven to possess powerful robustness [16, 17]. WavLM is a commonly used self-supervised pre-training module for extracting such representations [12]. For speaker embedding, we employ the SR module based on ECAPA-TDNN for offline extraction. Let $\mathbf{X}_{wav}$ is audio sequence, $\mathbf{X}_{mfcc}$ is mel frequency cepstrum coefficient (MFCC) features, and $\mathbf{Y}$ is text sequence. We utilize WavLM-Large to extract audio features $\mathbf{H}$, and the MFCC-based SR model to obtain speaker embedding $\mathbf{e}$. The process is described as:

$$\mathbf{H} = WavLM - Large(\mathbf{X}_{wav}) \quad (1)$$

$$\mathbf{E} = ECAPA - TDNN(\mathbf{X}_{mfcc}) \quad (2)$$

$$\mathbf{e} = SP(\mathbf{E}) \quad (3)$$

$$\mathbf{S} = Concat(Append(Linear_1(\mathbf{e}), T), Linear_2(\mathbf{H})) \quad (4)$$

$$\mathbf{F} = Conformer(\mathbf{S}) \quad (5)$$

$$\mathbf{P} = Decoder(\mathbf{F}, \mathbf{Y}), \quad (6)$$

where SSLR is $\mathbf{H} \in \mathbb{R}^{T\times d}$, T and d represent the number of frames and dimension. The MFCC-based SR model consists of two parts, Eq. (2) and (3). ECAPA-TDNN output is $\mathbf{E} \in \mathbb{R}^{n\times h}$, n and h representing output channels and dimension. Statistics pooling layer output is $\mathbf{e} \in \mathbb{R}^{1\times h}$. Eq. (4) combines the two features by concatenating them along feature dimension. $Linear_1(\mathbf{e}) \in \mathbb{R}^{1\times d'}$, $Linear_2(\mathbf{H}) \in \mathbb{R}^{T\times d'}$, $Append(Linear_1(\mathbf{e}), T) \in \mathbb{R}^{T\times d'}$, $\mathbf{S} \in \mathbb{R}^{T\times 2d'}$. $\mathbf{F}$ is the output of $Conformer(\cdot)$. $\mathbf{P}$ is posterior probability generated by $Decoder(\cdot)$ conditioned on $\mathbf{F}$ and $\mathbf{Y}$.

2.2. Proposed SAIS architecture

The SA-ASR mentioned above has limitations in enhancing the target speaker implicitly. These limitations arise from acoustic scenario mismatch, insufficient information in speaker embedding, and inadequate strength of embedding bias. To overcome these issues, we propose the SAIS framework which consists of three contributions: sharing SSLR, OT embedding, and HSGA.

Sharing SSLR. In SA-ASR, there is an issue of acoustic scenario mismatch between the ASR and SR modules, which includes mismatch of input features and spatial distribution. Specifically, input features for ASR and MFCC-based SR are $\mathbf{X}_{wav}$ and $\mathbf{X}_{mfcc}$. The x-vectors extracted offline come from a general speaker spatial distribution, which does not match spatial distribution of downstream ASR task. To address this issue, we adopt sharing SSLR strategy for both ASR and SR modules, as shown in Fig. 1 (b). For the issue of input features, ASR and SR modules share WavLM-Large feature extraction module (green box), as follows:

$$\hat{\mathbf{E}} = ECAPA{-}TDNN(\mathbf{H}) \quad (7)$$

$$\hat{\mathbf{e}} = SP(\hat{\mathbf{E}}), \quad (8)$$

where $\mathbf{H}$ is from Eq. (1). SSLR-based SR model involves Eq. (7) and (8). During SR model training stage, as shown in Fig. 1 (a), we freeze the parameters of WavLM-Large and train orange box part. The embedding $\hat{\mathbf{e}}$ is passed through output layer and AAM-Softmax [9] to obtain speaker posterior probabilities. Then, in Fig. 1 (b), we initialize and freeze green and gray boxes by trained SSLR-based SR model, and train ED ASR module (orange box) to obtain sharing SSLR-SR-ASR framework. During the above training of ED ASR and SR, WavLM-Large module (green box) is freezed, reducing input features differences between ASR and SR modules. To address the issue of spatial distribution mismatch, we apply JFT with a low learning rate on ASR training sets to reduce the gap in spatial distributions.

OT embedding. To obtain sufficient target speaker bias information, we introduce OT method to mitigate speaker information loss during generation of x-vectors through SP. This method involves defining a mapping matrix $\mathbf{M}$ and a cost matrix $\mathbf{C}$ to minimize information loss from $\hat{\mathbf{E}}$ to $\hat{\mathbf{e}}$. Let probability simplex $\mathbf{a}$ and $\mathbf{b}$ be the weights of the discrete

measures $\sum_i \mathbf{a}_i \delta_{\hat{\mathbf{E}}_i}$ and $\sum_j \mathbf{b}_j \delta_{\hat{\mathbf{e}}_j}$ with respective locations $\hat{\mathbf{E}}$ and $\hat{\mathbf{e}}$ [14], where δ is the Dirac at position $\hat{\mathbf{E}}$ or $\hat{\mathbf{e}}$. The entropic regularized Kantorovich relaxation [14] of OT:

$$\min_{\mathbf{M}\in U(\mathbf{a},\mathbf{b})} \sum_{ij} \mathbf{C}_{ij}\mathbf{M}_{ij} - \varepsilon H(\mathbf{M}), \tag{9}$$

where $H(\mathbf{M}) = -\sum_{ij} \mathbf{M}_{ij}(\log(\mathbf{M}_{ij}) - 1)$ is entropic regularization with parameter ε, U is space of admissible couplings. $\mathbf{M} \in \mathbb{R}_+^{n\times 1}$, $\mathbf{C} \in \mathbb{R}^{n\times 1}$. n and 1 is length of $\hat{\mathbf{E}}$, $\hat{\mathbf{e}}$. $\mathbf{M}$ and $\mathbf{a}$, $\mathbf{b}$ should satisfy the following relationship:

$$\mathbf{M1} = \mathbf{a}, \mathbf{M}^\top \mathbf{1}_n = \mathbf{b}, \tag{10}$$

where $\mathbf{1}_n$ is an n-dimensional identity matrix. Introducing two dual variables, $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{g} \in \mathbb{R}$, Eq. (9) is derived using Lagrange multiplier method:

$$\begin{aligned}\mathcal{L}(\mathbf{M},\mathbf{f},\mathbf{g}) &=< \mathbf{C}, \mathbf{M} > -\varepsilon H(\mathbf{M}) \\ &= - < \mathbf{f}, \mathbf{M1} - \mathbf{a} > - < \mathbf{g}, \mathbf{M}^\top \mathbf{1}_n - \mathbf{b} >\end{aligned} \tag{11}$$

$$\frac{\partial \mathcal{L}(\mathbf{M},\mathbf{f},\mathbf{g})}{\partial \mathbf{M}_{ij}} = \mathbf{C}_{i,j} + \varepsilon \log(\mathbf{M}_{i,j}) - \mathbf{f}_i - \mathbf{g}_j = 0. \tag{12}$$

Then, $\mathbf{M}$ is computed and can be rewritten in the form provided above using non-negative vectors $\mathbf{u}$, $\mathbf{v}$:

$$\mathbf{M}_{ij} = e^{\mathbf{f}_i/\varepsilon} e^{-\mathbf{C}_{i,j}/\varepsilon} e^{\mathbf{g}_j/\varepsilon} = \mathbf{u}_i \mathbf{K}_{i,j} \mathbf{v}_j. \tag{13}$$

Finally, these two updates define Sinkhorn's algorithm [14]:

$$\mathbf{u}^{(l+1)} = \frac{1/n}{\mathbf{K}\mathbf{v}^{(l)}}, \mathbf{v}^{(l+1)} = \frac{1}{\mathbf{K}^\top \mathbf{u}^{(l+1)}}, \tag{14}$$

where initialized with an arbitrary positive vector $\mathbf{v}^0 = \mathbf{1}$. $\mathbf{K} = \hat{\mathbf{E}}\mathbf{W}$, where $\mathbf{K} \in \mathbb{R}^{n\times 1}$, $\hat{\mathbf{E}} \in \mathbb{R}^{n\times h}$, $\mathbf{W} \in \mathbb{R}^{h\times 1}$. $\mathbf{W}$ is a trainable matrix, and it is minimized to reduce the loss of $\mathbf{C}$. l is the number of iterations. It is set to 10.

The calculation of $\mathbf{u}$ and $\mathbf{v}$ can be seen as expectation maximization (EM) algorithm. The E-step obtains $\mathbf{u}$ for the $l+1$ iteration by calculating $\mathbf{v}$ from the l-th iteration. The M-step, in the $l+1$-th iteration, updates $\mathbf{v}$ using $\mathbf{u}$. Then $\mathbf{M}$ is calculated by estimated $\mathbf{u}$, $\mathbf{v}$ and minimized $\mathbf{C}$ in Eq. (13). Finally, $\hat{\mathbf{E}}$ is mapped to $\hat{\mathbf{e}}$ by optimized $\mathbf{M}$.

HSGA. To enhance the strength of target speaker embedding bias, a novel approach called HSGA is proposed to optimize the feature fusion layer in Fig. 1 (b), which is applied to each layer of conformer in Fig. 1 (c). Feature fusion layer can be refered in Eq. (4). Specifically as follows:

$$\mathbf{Q}_m = DNN(\hat{\mathbf{H}}_{m-1}) \tag{15}$$

$$\mathbf{K}_m, \mathbf{V}_m = Append(\sigma(Linear_1(\hat{\mathbf{e}})), T) \odot \mathbf{Q}_m \tag{16}$$

$$\mathbf{S}_m = SelfAttn(\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m) \tag{17}$$

$$\hat{\mathbf{H}}_m = DNN(ConvBlock(\mathbf{S}_m)), \tag{18}$$

where $\hat{\mathbf{H}}_m$ represents the output of each layer of conformer, M denotes the number of conformer layers. $\hat{\mathbf{H}}_0 = Linear_2(\mathbf{H})$. $\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m \in \mathbb{R}^{T\times d}$. $\odot$ is element-wise multiplication. σ represents $sigmoid$ function. Eq. (16) can control speaker information within the range of $(0, 1)$, and scale inputs of different conformer layers. In those different layers, lower layers contain speaker information, higher layers contain semantic information [11]. For the scaled results, $\mathbf{K}_m, \mathbf{V}_m$, the important frequency components are enhanced while the less important ones are attenuated. This ensures that during the encoder network from low-level to high-level representations, sufficient target speaker bias information is incorporated, allowing ASR model to better distinguish target speaker corresponding to semantic units.

3. EXPERIMENTS

3.1. Experimental setups

We primarily focus on DASR task in CHiME-7 challenge. In this task, sub-track allows the use of oracle diarization, which can better demonstrate impact by ASR model. SSLR-SR model is trained on the VoxCeleb1&2 [18] and its 3-fold data augmentation with available MUSAN (babble, noise, music) [19]. ASR training sets are described as shown in Table 1. For 470 hours data, CHiME-6 and Mixer 6 are from official baseline training sets [17]. They are augmented using same method as [17], including 3-fold speed perturbation. In addition to the 470 hours, LibriSpeech [20] and weakly labeled VoxCeleb1&2 are simulated with added noise and reverberation as additional training data [19, 21]. The only dev sets (CHiME-6, DiPCo, Mixer 6) were used for testing due to lack of labeled evaluation sets [17]. Word error rate (WER) and relative WER reduction (WERR) are employed as evaluation metrics.

Table 1. Statistics of ASR training sets.

Duration(h)	Corpus	Sample Scale
470	CHiME-6 (GSS, near), Mixer 6 (near)	x3
1400	470 hours + LibriSpeech (simu)	x1
3800	1400 hours + VoxCeleb 1&2 (simu)	x1

In proposed framework, WavLM-Large module contains 24-layers transformer and 316M parameters as described in [12]. The SR module based on ECAPA-TDNN apply 512 channels as described in [9] and get 192 dimensions x-vector. The ASR module contains 12 encoder conformer layers and 6 decoder transformer layers as described in [16].

3.2. Results and analysis

Table 2 summarizes SAIS and state-of-the-art (SOTA) results. E1 is Whisper results from official testing [17]. E2 is CHiME-7 official baseline [17]. E3 represents SSLR-based ED ASR resluts. It has a similar structure and training sets as E2. E4 uses Fbank features as input instead of SSLR in E3. E5 is SA-ASR model described in section 2.1. It incorporates pretrained MFCC-based ECAPA-TDNN [22] for offline x-vector extraction as speaker adaptation. E6 represents the proposed SAIS. From the last column, SSLR (E3) has significantly improvement compared to Fbank (E4) due to its richer prior knowledge. The proposed SAIS (E6) achieves 17.53% relative improvement in WERR compared with Whisper (E1).

Table 2. WER results of SAIS method and current SOTA on the development set of CHiME-7.

ID	Model	CHiME-6	DiPCo	Mixer 6	Ave	WERR
E1	Whisper [17]	30.90	34.50	21.20	28.80	—
E2	baseline [17]	32.60	33.50	20.20	28.80	0.00
E3	SSLR-ASR	31.66	34.46	17.86	27.99	2.81
E4	Fbank ASR	57.07	45.76	68.64	57.16	-98.47
E5	SA-SSLR-ASR	31.21	34.19	17.53	27.64	4.03
E6	SAIS (470 h)	25.74	29.66	15.85	23.75	17.53

Table 3. Analysis of SAIS method through ablation studies.

ID	Model	Ave	WERR
E1	Whisper [17]	28.80	—
E7	Sharing SSLR+SP+Concat	25.63	11.01
E8	E7 w/o JFT	26.54	7.85
E9	Sharing SSLR+OT+Concat	24.77	13.99
E10	SAIS (HSGA 6 layers)	24.13	16.22
E6	SAIS (HSGA 12 layers)	23.75	17.53

We analyze the reasons behind significant gains obtained by SAIS in Table 3. E7 represents sharing SSLR for both SR and ASR. E8 lacks JFT strategy comparing with E7. E9 optimizes x-vector generation from SP to OT. E10 applies HSGA to only half of encoder layers. We observe that sharing SSLR (E7) achieves the highest gains, and without using JFT (E8), the gains decrease significantly due to acoustic mismatch. OT embedding (E9) provides 3.35% relative improvement in WERR compared with SP (E7) due to more informative embedding. Applying HSGA to entire encoder (E6) has higher gains than E10 due to stronger target speaker bias.

We analyze implicit speaker enhancement of SAIS in Table 4. We focus on the recognition results of E5 and E6 for CHiME-6 dev sets. It have S02 and S09 scenario. Since each scenario consists of four speakers, N_spk, $N \in (1, 2, 3)$ indicates the number of non-target speakers in current speech segment, excluding the target speaker. The first row represents the percentage of total duration of non-target speakers in the current segment, reflecting the degree of overlapping. The last three rows indicate relative improvement of E6 compared to E5. In every row, SAIS achieves better implicit speaker enhancement performance as the degree of overlapping increases. In every column, the benefits of SAIS diminish significantly as the number of speakers increases. However, E6 maintains a lower WER compared to E5.

Table 5 shows the impact of different data sizes and structures on SAIS. E11 and E14 represent the results trained using 1400h and 3800h of Table 1, respectively. The weak labels for VoxCeleb1&2 are generated from E11. E12 and E13 involve expanding 12-layer encoder of E11 from 256 units to 384 and 512 units, respectively, but performance degradation is observed on 1400h training sets. Inspired by [16], we also introduce Conv-TasNet as speech enhancement (SE) module in the SAIS frontend to improve noise robustness [16]. This SE model is pretrained on LibriSpeech and MUSAN. E15 and E16 represent fine-tuning SAIS with fixed SE parameters on different training sets. We also employ JFT strategy on entire framework in E17, achieving 28.19% relative improvement in WERR compared with Whisper. E18 is the USTC-NERCSLIP fusion system with four ASR models [23]. Among them, all models are optimized and trained based on SAIS. This system achieves top rankings in CHiME-7 DASR.

Table 4. Analysis of implicit enhancement of target speaker in segments with varying speaker overlap on CHiME-6 DEV.

S02/S09	overlap	0-50%	50-100%	100-150%	150-200%
E5	1_spk	23.89	43.83	—	—
	2_spk	17.83	28.79	49.56	62.03
	3_spk	17.98	23.93	34.44	53.68
E6	1_spk	19.54	33.55	—	—
	2_spk	14.86	23.07	38.38	46.51
	3_spk	16.86	20.49	27.86	42.43
WERR (E5/E6)	1_spk	18.21	23.45	—	—
	2_spk	16.66	19.87	22.56	25.02
	3_spk	6.23	14.38	19.11	20.96

Table 5. WER results on CHiME-7 development set for different training sets and model architectures.

ID	Model	CHiME-6	DiPCo	Mixer 6	Ave	WERR
E1	Whisper [17]	30.90	34.50	21.20	28.80	—
E11	SAIS (1400 h)	25.56	29.19	15.37	23.37	18.85
E12	E11 (12x384)	26.63	30.08	17.16	24.29	15.66
E13	E11 (12x512)	29.20	31.80	16.88	25.96	9.86
E14	SAIS (3800 h)	24.28	29.09	14.45	22.61	21.49
E15	SE + E11	24.66	28.82	14.76	22.75	21.01
E16	SE + E14	24.75	27.77	13.43	21.98	23.68
E17	SE + E14 (JFT)	22.27	26.94	12.84	20.68	28.19
E18	USTC-NERCSLIP [23]	19.60	24.10	12.20	18.60	35.42

4. CONCLUSIONS

This work aims to enhance the capability of implicit target speaker enhancement in ASR. To achieve this, we propose an efficient SAIS framework that optimizes joint SR and ASR models for speaker-adaptive speech recognition. It contributes in three aspects: 1) introducing SSLR for ASR input features and online x-vector extraction, ensuring consistency in acoustic scenarios of the two types of features. 2) optimizing x-vector generation from SP to OT module, obtaining more informative embedding. 3) proposing HSGA module, applied in different encoders of ASR, to ensure sufficient target speaker bias from low-level to high-level representations. In experiments, proposed method achieves 28.19% relative improvement in WERR compared with official baseline on dev sets. We observe implicit speaker enhancement capability in our approach more as the degree of speech overlap increases. In CHiME-7 DASR, our approach also ensures the USTC-NERCSLIP fusion system achieves top rankings.

5. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant 62171427.

6. REFERENCES

[1] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal, "The fifth'chime'speech separation and recognition challenge: dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[4] Shinji Watanabe, Michael Mandel, Jon Barker, and Emmanuel Vincent, "Overview of the 6th chime challenge," in *CHiME6 Workshop*, 2020.

[5] Christoph Boeddeker, Jens Heitkaemper, Joerg Schmalenstroeer, et al., "Front-end processing for the chime-5 dinner party scenario," in *CHiME5 Workshop, Hyderabad, India*, 2018, vol. 1.

[6] Desh Raj, Daniel Povey, and Sanjeev Khudanpur, "Gpu-accelerated guided source separation for meeting transcription," *arXiv preprint arXiv:2212.05271*, 2022.

[7] Felix Weninger, Jesús Andrés-Ferrer, Xinwei Li, and Puming Zhan, "Listen, attend, spell and adapt: Speaker adapted sequence-to-sequence asr," *arXiv preprint arXiv:1907.04916*, 2019.

[8] Vishwas M Shetty, S Umesh, et al., "Investigation of speaker-adaptation methods in transformer based asr," *arXiv preprint arXiv:2008.03247*, 2020.

[9] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[10] Christoph Lüscher, Jingjing Xu, Mohammad Zeineldeen, Ralf Schlüter, and Hermann Ney, "Improving and analyzing neural speaker embeddings for asr," *arXiv preprint arXiv:2301.04571*, 2023.

[11] Haitao Tang, Yu Fu, Lei Sun, Jiabin Xue, et al., "Reducing the gap between streaming and non-streaming transducer-based asr by adaptive two-stage knowledge distillation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[12] Sanyuan Chen, Chengyi Wang, Zhengyang 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.

[13] Murali Karthick Baskar, Tim Herzig, Diana Nguyen, et al., "Speaker adaptation for wav2vec2 based dysarthric asr," *arXiv preprint arXiv:2204.00770*, 2022.

[14] Gabriel Peyré, Marco Cuturi, et al., "Computational optimal transport," *Center for Research in Economics and Statistics Working Papers*, , no. 2017-86, 2017.

[15] Anmol Gulati, James Qin, Chung-Cheng Chiu, et al., "Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[16] Xuankai Chang, Takashi Maekaku, Yuya Fujita, and Shinji Watanabe, "End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation," 2022.

[17] Samuele Cornell, Matthew Wiesner, Shinji Watanabe, et al., "The chime-7 dasr challenge: Distant meeting transcription with multiple devices in diverse scenarios," *arXiv preprint arXiv:2306.13734*, 2023.

[18] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 67–74.

[19] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] Keisuke Kinoshita, Marc Delcroix, et al., "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[22] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, et al., "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[23] Ruoyu Wang, Maokui He, Jun Du, et al., "The ustc-nercslip systems for the chime-7 dasr challenge," *arXiv preprint arXiv:2308.14638*, 2023.