A PRACTICAL TWO-STAGE TRAINING STRATEGY FOR MULTI-STREAM END-TO-END SPEECH RECOGNITION

Ruizhi Li[1], *Gregory Sell*[1,2], *Xiaofei Wang*[3*], *Shinji Watanabe*[1], *Hynek Hermansky*[1,2]

[1]Center for Language and Speech Processing, The Johns Hopkins University, USA
[2]Human Language Technology Center of Excellence, The Johns Hopkins University, USA
[3]Speech and Dialog Research Group, Microsoft, USA

ABSTRACT

The multi-stream paradigm of audio processing, in which several sources are simultaneously considered, has been an active research area for information fusion. Our previous study offered a promising direction within end-to-end automatic speech recognition, where parallel encoders aim to capture diverse information followed by a stream-level fusion based on attention mechanisms to combine the different views. However, with an increasing number of streams resulting in an increasing number of encoders, the previous approach could require substantial memory and massive amounts of parallel data for joint training. In this work, we propose a practical two-stage training scheme. Stage-1 is to train a Universal Feature Extractor (UFE), where encoder outputs are produced from a single-stream model trained with all data. Stage-2 formulates a multi-stream scheme intending to solely train the attention fusion module using the UFE features and pretrained components from Stage-1. Experiments have been conducted on two datasets, DIRHA and AMI, as a multi-stream scenario. Compared with our previous method, this strategy achieves relative word error rate reductions of 8.2–32.4%, while consistently outperforming several conventional combination methods.

Index Terms**—** End-to-End Speech Recognition, Multi-Stream, Multiple Microphone Array, Two-Stage Training

1. INTRODUCTION

The multi-stream paradigm in speech processing considers scenarios where parallel streams carry diverse or complementary task-related knowledge. In these cases, an appropriate strategy to fuse streams or select the most informative source is necessary. One potential source of inspiration in this setting is from the observations of parallel processing in the human auditory system, and resulting innovations have been successfully applied to conventional automatic speech recognition (ASR) frameworks [1, 2, 3, 4]. For instance, multi-band acoustic modeling was formulated to address noise robustness [1, 3]. [5] investigated several performance measures in spatial acoustic scenes to choose the most reliable source for hearing aids. The multi-modal applications combine visual [6] or symbolic [7] inputs together with audio signal to improve speech recognition.

The work that follows considers far-field ASR using multiple microphone arrays, a specific case of multi-stream paradigm. Without any knowledge of speaker-array distance or orientation, it is still challenging to speculate which array is most informative or least corrupted. The common methods of utilizing multiple arrays in conventional ASR are posterior combination [8, 9], ROVER [10], distributed beamformer [11], and selection based on Signal-to-Noise/Interference Ratio (SNR/SIR) [12].

In recent years, with the increasing use of Deep Neural Networks (DNNs) in ASR, End-to-End (E2E) speech recognition approaches, which directly transcribe human speech into text, have received greater attention. The E2E models combine several disjoint components (acoustic model, pronunciation model, language model) from hybrid ASR into one single DNN for joint training. Three dominant E2E architectures for ASR are Connentionist Temporal Classification (CTC) [13, 14, 15], attention-based encoder decoder [16, 17], and Recurrent Neural Network Transducer (RNN-T) [18, 19]. Coupled with a CTC network within a multi-task scheme, the joint CTC/Attention model [20, 21, 22] outperforms the attention-based model by addressing misalignment issues, achieving the state-of-the-art E2E performance on several benchmark datasets [22].

In [23], we proposed a novel multi-stream model based on a joint CTC/Attention E2E scheme, where each stream is characterized by a separate encoder and CTC network. A Hierarchical Attention Network (HAN) [24, 25] acts as a fusion component to dynamically assign higher weights for streams carrying more discriminative information for prediction. The Multi-Encoder Multi-Array (MEM-Array) framework was introduced in [23] to improve the robustness of distant microphone arrays, where each array is represented by a separate encoder. While substantial improvements were reported within a two-stream configuration, there are two concerns when more streams are involved. First, during training, fitting all parallel encoders in device computing memory is potentially impractical for joint optimization, as the encoder is typically the largest component by far, i.e., 88% of total parameters in this work. Second, due to the data-hungry nature of DNNs and the expensive cost of collecting parallel data, training multiple models with excess degrees of freedom is not optimal.

In this paper, we present a practical two-stage training strategy on the MEM-Array framework targeting the aforementioned issues. The proposed technique has the following highlights:

1. In Stage-1, a single-stream model is trained using all data for better model generalization. The encoder will then acts as a Universal Feature Extractor (UFE) to process parallel data individually to generate a set of high-level parallel features.
2. Initializing components (CTC, decoder, frame-level attention) from Stage-1, Stage-2 training only optimizes the HAN component operating directly on UFE parallel features. The resulting memory and computation savings greatly simplify training, potentially allowing for more hyperparameter exploration or consideration of more complicated architectures.

[*] This work was done while Xiaofei Wang was a post-doc at Johns Hopkins University.

3. Lack of adequate volume of data, specially parallel data, leads to overfit or is hard to tackle unseen data. The proposed two-stage strategy better defines the data augmentation scheme. Augmentation in Stage-1 aims to extract more discriminative high-level features and provides well-pretrained modules for Stage-2, whereas Stage-2 could focus on improving the robustness of information fusion.

2. MEM-ARRAY END-TO-END SPEECH RECOGNITION

In this section, we review the joint CTC/Attention framework and the extended MEM-Array model, one realization of the multi-stream approach with focus on distant multi-array scenario.

2.1. Joint CTC/Attention

The joint CTC/Attention architecture, illustrated in Stage-1 of Fig. 1, takes advantage of both CTC and attention-based models within a Multi-Task Learning (MTL) scheme. The model directly maps a T-length sequence of D-dimensional speech vectors, $X = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, 2, ..., T\}$, into an L-length label sequence, $C = \{c_l \in \mathcal{U} | l = 1, 2, ..., L\}$. Here $\mathcal{U}$ is a set of distinct labels. The encoder transforms the acoustic sequence X into a higher-level feature representation $H = \{\mathbf{h}_1, ..., \mathbf{h}_{\lfloor T/s \rfloor}\}$, which is shared for the use of CTC and attention-based models. Here, $\lfloor T/s \rfloor$ time instances are generated at the encoder-output level with a subsampling factor of s. The loss function to be optimized is a logarithmic linear combination of CTC and attention objectives, i.e., $p_{\text{ctc}}(C|X)$ and $p^{\dagger}_{\text{att}}(C|X)$:

$$\mathcal{L}_{\text{MTL}} = \lambda \log p_{\text{ctc}}(C|X) + (1 - \lambda) \log p^{\dagger}_{\text{att}}(C|X), \tag{1}$$

where $\lambda \in [0, 1]$ is a hyper parameter. Note that $p^{\dagger}_{\text{att}}(C|X)$ is an approximated letter-wise objective where the probability of a prediction is conditioned on previous true labels. During inference, a label-synchronous beam search is employed to predict the most probable label sequence $\hat{C}$:

$$\begin{aligned}\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{&\lambda \log p_{\text{ctc}}(C|X) + (1 - \lambda) \log p_{\text{att}}(C|X) \\ &+ \gamma \log p_{\text{lm}}(C)\},\end{aligned} \tag{2}$$

where $\log p_{\text{lm}}(C)$ is evaluated from an external Recurrent Neural Network Language Model (RNN-LM) with a scaling factor γ.

2.2. MEM-Array Model

An end-to-end ASR model addressing the general multi-stream setting was introduced in [23]. As one representative framework, MEM-Array concentrates on cases of far-field microphone arrays to handle different dynamics of streams. The architecture of N streams is shown in Stage-2 of Fig. 1. Each encoder operates separately on a parallel input $X^{(i)}$ to extract a set of frame-wise hidden vectors $H^{(i)}$:

$$H^{(i)} = \text{Encoder}^{(i)}(X^{(i)}), i \in \{1, ..., N\}, \tag{3}$$

where we denote superscript i as the index for stream i, and $H^{(i)} = \{\mathbf{h}^{(i)}_1, ..., \mathbf{h}^{(i)}_{\lfloor T^{(i)}/s \rfloor}\}$. A frame-level attention mechanism is designated to each encoder to carry out the stream-specific speech-label alignment. For stream i, the letter-wise context vector $\mathbf{r}^{(i)}_l$ is computed via a location-based attention network [26] as follows:

$$\mathbf{r}^{(i)}_l = \sum_{t=1}^{\lfloor T^{(i)}/s^{(i)} \rfloor} a^{(i)}_{lt} \mathbf{h}^{(i)}_t, \tag{4}$$

$$a^{(i)}_{lt} = \text{Attention}(\{a^{(i)}_{l-1}\}^{T^{(i)}}_{t=1}, \mathbf{q}_{l-1}, \mathbf{h}^{(i)}_t), \tag{5}$$

where $a^{(i)}_{lt}$ is the attention weight, a soft-alignment of $\mathbf{h}^{(i)}_t$ for output c_l, and $\mathbf{q}_{l-1}$ is the previous decoder state. In the multi-stream setting, the contribution of each stream changes dynamically. Hence, a secondary stream attention, the HAN component, is exploited for the purpose of robustness. The fusion context vector $\mathbf{r}_l$ is obtained as a weighted summation of $\{\mathbf{r}^{(i)}_l\}^N_{i=1}$:

$$\mathbf{r}_l = \sum_{i=1}^{N} \beta^{(i)}_l \mathbf{r}^{(i)}_l, \tag{6}$$

$$\beta^{(i)}_l = \text{HierarchicalAttention}(\mathbf{q}_{l-1}, \mathbf{r}^{(i)}_l), i \in \{1, ..., N\}. \tag{7}$$

where in this work, a content-based attention network [26] is applied here, and $\beta^{(i)}_l$ is a Softmax output across $\{i\}^N_1$ from the HAN component, a stream-level attention weight for array i of prediction c_l. In addition, a separate CTC network is active for each encoder to enhance the stream diversity instead of sharing a CTC across all streams. In this setting, the MEM-Array model follows Eq. (1) and (2) with a modified CTC objective:

$$\log p_{\text{ctc}}(C|X) = \frac{1}{N} \sum_{i=1}^{N} \log p_{\text{ctc}^{(i)}}(C|X), \tag{8}$$

where joint CTC loss is the average of per-encoder CTCs.

3. PROPOSED TRAINING STRATEGY

In this section, we present a practical two-stage training strategy for the MEM-Array model, depicted in Fig. 1. The details of each stage are discussed in the following sections.

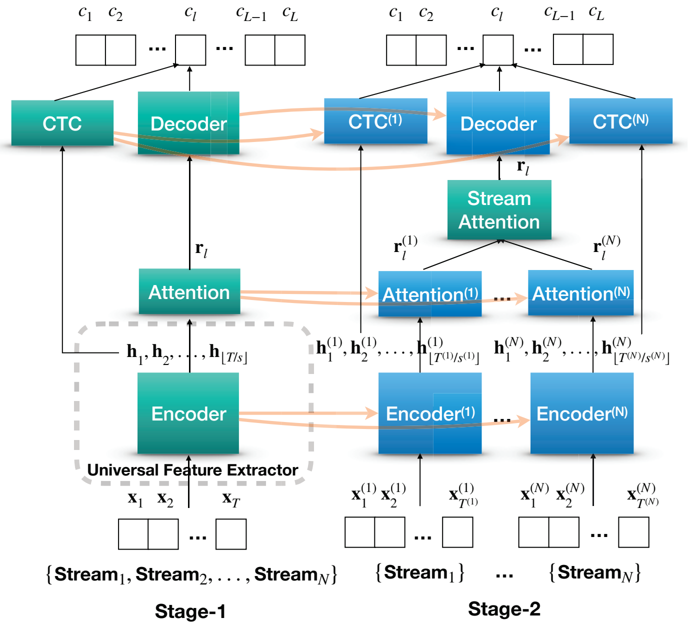
c_1 c_2 c_l c_{L-1} c_L
CTC
Decoder
CTC(1)
Decoder
CTC(N)
$\mathbf{r}_l$
Stream Attention
$\mathbf{r}_l^{(1)}$
$\mathbf{r}_l^{(N)}$
$\mathbf{r}_l$
Attention
Attention(1)
Attention(N)
$\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{\lfloor T/s \rfloor}$
$\mathbf{h}^{(1)}_1, \mathbf{h}^{(1)}_2, \ldots, \mathbf{h}^{(1)}_{\lfloor T^{(1)}/s^{(1)} \rfloor}$
$\mathbf{h}^{(N)}_1, \mathbf{h}^{(N)}_2, \ldots, \mathbf{h}^{(N)}_{\lfloor T^{(N)}/s^{(N)} \rfloor}$
Encoder
Encoder(1)
Encoder(N)
Universal Feature Extractor
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_T$
$\mathbf{x}^{(1)}_1$ $\mathbf{x}^{(1)}_2$ $\mathbf{x}^{(1)}_{T^{(1)}}$
$\mathbf{x}^{(N)}_1$ $\mathbf{x}^{(N)}_2$ $\mathbf{x}^{(N)}_{T^{(N)}}$
$\{\text{Stream}_1, \text{Stream}_2, \ldots, \text{Stream}_N\}$
$\{\text{Stream}_1\}$
$\{\text{Stream}_N\}$
Stage-1
Stage-2

Fig. 1: Proposed Two-Stage Training Strategy. Color "green" indicates the components are trainable; Color "blue" means parameters of the components are frozen.

3.1. Stage 1: Universal Feature Extractor

The intent of Stage-1 is to obtain a single well-trained encoder, which we refer to as Universal Feature Extractor (UFE), to prepare a new set of high-level features for Stage-2. Encoder in E2E model can be viewed as an acoustic modeling that generates sequences $H = \{\mathbf{h}_1, ..., \mathbf{h}_{\lfloor T/s \rfloor}\}$ with more discriminative power for prediction. We denote the encoder outputs H as the UFE features. In general, the majority of the overall parameters are contained in the encoder.

In Stage-1, a single-stream joint CTC/Attention model is optimized as shown in Fig. 1. Audio features from all available streams are used to train the model. After training, we extract UFE features $H^{(i)} = \{\mathbf{h}_1^{(i)}, ..., \mathbf{h}^{(i)}_{\lfloor T^{(i)}/s \rfloor}\}$ for each stream i, separately. Since subsampling mitigates the increased dimension of UFE features, it is possible to save the UFE features at a similar size to the original speech features. Moreover, byproducts in Stage-1, such as decoder, CTC and attention, can be used for initialization in Stage-2.

3.2. Stage 2: Parallel-Stream Fusion

As illustrated in Fig. 1, Stage-2 focuses on training the fusion component within the multi-stream context. The MEM-Array model uses parallel encoders as information streams. The previous strategy uses joint training with multiple large encoders, which is expensive in memory and time for more complex models or more streams. Taking advantage of UFE features greatly alleviates this complication.

In Stage-2, we formulate a multi-stream scheme on UFE features $\{H^{(i)}\}_{i=1}^{N}$ as parallel inputs. In this model, parameters of all components, except the stream attention module, are initialized from Stage-1 and frozen during optimization. The stream fusion component is randomly initialized, and is the only trainable element in Stage-2. Without any involvement of encoders, frame-level attention directly operates on UFE features. This setup not only reduces the amount of required parallel data, but it also greatly reduces memory and time requirements, allowing for more thorough hyperparameter exploration or utilization of more complex architectures.

4. EXPERIMENTAL SETUP

We demonstrated the two-stage training strategy using two datasets: DIHRA English WSJ [27] and AMI Meeting Corpus [28].

The DIRHA English WSJ is part of the DIRHA project, which focuses on speech interaction in domestic scenes via distant microphones. There are in total 32 microphones placed in an apartment setting with a living room and a kitchen. We chose a 6-mic circular array (Beam Circular Array) and an 11-mic linear array (Beam Linear Array) in the living room for experiments with two parallel streams. Additionally, a single microphone (L1C) was picked to serve as a third stream in 3-stream experiments. Training data was created by contaminating original Wall Street Journal data (WSJ0 and WSJ1, 81 hours per stream), providing room impulse responses for each stream. Simulated WSJ recordings with domestic background noise and reverberation were used as the development set for cross validation. The evaluation set has 409 WSJ recordings uttered in real domestic scenario.

The AMI Meeting Corpus was collected in three instrumented rooms with meeting conversations. Each room has two microphone arrays to collect 100 hours of far-field signal-synchronized recordings. With no speakers overlapping, the training, development and evaluation set have 81 hours, 9 hours and 9 hours of meeting recordings, respectively. No close-talk microphone recordings are used here.

We designed both 2-stream and 3-stream settings for DIRHA and 2-stream experiments for AMI. Note that for each array, the multi-channel input was synthesized into single-channel audio using Delay-and-Sum beamforming with BeamformIt [29]. Experiments were conducted using a Pytorch back-end on ESPnet [30] configured as described in Table 1.

Table 1: Experimental Configuration.

Feature	80-dim log-mel filter bank + 3-dim pitch
Model	
Encoder type	VGGBLSTM [21, 31] (subsampling factor: 4)
Encoder layers	6(CNN)+2(BLSTM)
Encoder units	320 cells (BLSTM layers)
Encoder projection	320 cells (BLSTM layers)
Frame-level Attention	320-cell Content-based
Stream Attention	320-cell Location-based
Decoder type	1-layer 300-cell LSTM
Train and Decode	
Optimizer	AdaDelta (Batch size: 15)
Training Epoch	30 epochs (patience:3 epochs)
CTC weight λ	0.2 (train); 0.3 (decode)
Label Smoothing	Type: Unigram [32], Weight: 0.05
RNN-LM	
Type	Look-ahead Word-level RNNLM [33]
Train data	AMI:AMI; DIRHA:WSJ0-1+extra WSJ text
LM weight γ	AMI:0.5; DIRHA:1.0

5. RESULTS AND DISCUSSIONS

Firstly, we examined UFE features in a single-stream setting. Next, the full proposed strategy was analyzed in comparison to the previous approach as well as to several conventional fusion methods on DIRHA 2-stream case. Results on AMI and extension with more streams on DIRHA were explored as well. Lastly, we considered the potential benefits of data augmentation in this framework.

5.1. Effectiveness of Two-Stage Training

In this section, we discuss the results on 2-stream DIRHA to demonstrate the value of proposed strategy. First, to evaluate Stage-1 training, Character/Word Error Rates (CER/WER) results on single stream systems are summarized in Table 2. Training the model using data from both streams improves performance substantially on the individual arrays, i.e., $37.6\% \rightarrow 33.9\%$ and $39.2\% \rightarrow 30.7\%$. The UFE features are the outputs of an encoder trained with this improved strategy. In our setup, 320-dimensional UFE features took slightly smaller space than 83-dimensional acoustic frames since the subsampling factor $s = 4$.

Table 2: Stage-1 results on 2-stream DIRHA.

	Arr_1		Arr_2	
Train Data	CER(%)	WER(%)	CER(%)	WER(%)
Single Stream				
Arr_1	22.3	37.6	–	–
Arr_2	–	–	23.0	39.2
$\text{Arr}_1, \text{Arr}_2$	**20.1**	**33.9**	**17.9**	**30.7**

Table 3 illustrates several training strategies in Stage-2. Since Stage-2 operates on UFE features directly, its training only involves, at most, frame-level attention (ATT), decoder (DEC), hierarchical attention (HAN) and CTC. These experiments considered which of these components should be initialized from their Stage-1 counterparts, as well as which components should be fine-tuned or frozen during Stage-2 updates. In both cases of fine-tuning or freezing Pre-Trained (PT) modules in Stage-2, more noticeable improvements were reported with introducing more pretraining knowledge, i.e., 32.9% → 28.4% and 31.8% → 26.8%, respectively. Moreover, keeping all PT components frozen during Stage-2 and training solely the fusion module showed relative WER reduction of 5.6% (28.4% → 26.8%) with only 0.2 million active parameters. Overall, a substantial improvement of 18.8% relative WER reduction (33.0% → 26.8%) was observed compared to jointly training a massive model, including encoders, from scratch.

Table 3: WER(%) Comparison among various Stage-2 training strategies on 2-stream DIRHA. Note that components with random initialization in Stage-2 are listed in parentheses of first column. The amount of trainable parameters in Stage-2 when freezing Stage-1 Pre-Trained (PT) components is stated in parentheses of last column.

Initialization with PT Comp. (rand. init. comp.)	Fine-tune PT Comp.	Freeze PT Comp.
No Two-Stage		
Baseline	–	33.0 (21.82M)
Two-Stage		
– (ATT, DEC, CTC, HAN)	32.9	31.8 (1.78M)
CTC (ATT, DEC, HAN)	34.4	30.7 (1.75M)
ATT (DEC, CTC, HAN)	33.3	30.6 (1.37M)
ATT, DEC (CTC, HAN)	29.0	27.4 (0.23M)
ATT, DEC, CTC (HAN)	**28.4**	**26.8** (0.20M)

5.2. Multi-Stream v.s. Conventional Methods

In Table 4, the MEM-Array model with our two-stage training strategy consistently outperforms the baseline model which needs joint training after random initialization. 18.8%, 32.4%, and 8.2% relative WER reductions are achieved in 2-stream DIRHA, 3-stream DIRHA, and 2-stream AMI, respectively. Note that AMI experients were conducted using VGGBLSTM with 2-layer BLSTM layers without any close-talk recordings and data perturbations. It is worth mentioning that those reductions in WERs were accomplished while simultaneously significantly decreasing the number of unique parameters in training by avoiding costly multiples of the large encoder component (10 million parameters per stream, in this case).

In addition, results from several conventional fusion strategies are shown in Table 4: signal-level fusion via WAV alignment and average; feature-level frame-by-frame concatenation; word-level prediction fusion using ROVER. For fair comparison, single-level and word-level fusion models utilized Stage-1 pre-trained models as their initialization. Note that word-level fusion operates on decoding results from pretrained single-stream from Stage-1. Still, our proposed strategy consistently performs better than all other fusion methods in all conditions.

5.3. Discussion on Data Augmentation

The two-stage training strategy provides various opportunities for data augmentation. Stage-1 does not consider parallel data, so any augmentation technique for regular E2E ASR could be applied in this stage to improve the robustness of the UFE. Stage-2 augmentation, on the other hand, would be expected to improve robustness of the combination of corrupted individual streams. In this study, we employed a simple data augmentation technique called SpecAugment [34], which randomly removes sections of the input signal in a structured fashion, to demonstrate the potential of this direction. Table 5 shows the improvements from applying SpecAugment on two separate training stages. The best performance was from data augmentation on Stage-1 when freezing all Stage-1 pretrained components. With additional Stage-2 SpecAugment, there was not a noticeable difference in terms of WERs (22.6% v.s. 22.4% and 22.6% v.s. 22.5%). 10% absolute WER reduction was achived in AMI with two stage augmentation. However, it is important to remember that, while the performance gap from fine-tuning versus freezing pre-trained components is narrowed with Stage-2 augmentation, the reductions in Stage-2 memory and computation requirements are still substantially better with frozen parameters.

Table 4: WER(%) Comparson between proposed two-stage approach and alternative conventional methods.

	Unique	# Streams		
	Params	DIRHA		AMI
Model	(in million)	2	3	2
MEM-Array Model				
Baseline [23]	21.8(2),32.1(3)	33.0	32.1	59.5
Proposed Strategy	11.6	**26.8**	**21.7**	**54.6**
Other Fusion Methods				
WAV Align.& Avg.	11.4	32.4	30.1	55.9
Frame Concat.	16.9(2),23.8(3)	33.7	33.8	59.4
ROVER	11.4	34.2	23.6	58.0

Table 5: Performance (WER(%)) investigation of two-stage data augmentation using SpecAugment on 2-stream DIRHA and AMI.

	DIHRA		
	Fine-tune	Freeze	
Model	PT Comp.	PT Comp.	AMI
Augmentation			
no SpecAugment	28.4	26.8	59.5
Stage-1	22.6	**22.4**	55.8
Stage-1, Stage-2	22.5	22.6	**49.2**

6. CONCLUSIONS

In this work, we proposed a practical two-stage training strategy to improve multi-stream end-to-end ASR. A universal feature extractor is trained in Stage-1 with all available data. In Stage-2, a set of high-level UFE features are used to train a multi-stream model without requiring highly-parameterized parallel encoders. This two-stage strategy remarkably alleviates the burden of optimizing a massive multi-encoder model while still substantially improving the ASR performance. This work shows great potential and value for this approach, but numerous directions remain for future exploration. More sophisticated data augmentation techniques beyond the single method considered here should be explored. Stage-2 training could also possibly benefit from stream-specific knowledge by exploiting more complex stream attention. Strategies for adding new streams to an existing model would also be worth investigating.

7. REFERENCES

[1] Sri Harish Reddy Mallidi, *A Practical and Efficient Multistream Framework for Noise Robust Speech Recognition*, Ph.D. thesis, Johns Hopkins University, 2018.

[2] Hynek Hermansky, "Multistream recognition of speech: Dealing with unknown unknowns," *Proc. of the IEEE*, vol. 101, no. 5, pp. 1076–1088, 2013.

[3] Sri Harish Mallidi and Hynek Hermansky, "Novel neural network based fusion for multistream asr," in *Proc. of ICASSP*. IEEE, 2016, pp. 5680–5684.

[4] Hynek Hermansky, "Coding and decoding of messages in human speech communication: Implications for machine recognition of speech," *Speech Communication*, 2018.

[5] Bernd T Meyer, Sri Harish Mallidi, Angel Mario Castro Martinez, Guillermo Payá-Vayá, Hendrik Kayser, and Hynek Hermansky, "Performance monitoring for automatic speech recognition in noisy multi-channel environments," in *Proc. of SLT*. IEEE, 2016, pp. 50–56.

[6] Shruti Palaskar, Ramon Sanabria, and Florian Metze, "End-to-end multimodal speech recognition," in *Proc. of ICASSP*. IEEE, 2018, pp. 5774–5778.

[7] Adithya Renduchintala, Shuoyang Ding, Matthew Wiesner, and Shinji Watanabe, "Multi-modal data augmentation for end-to-end asr," *Proc. Interspeech 2018*, pp. 2394–2398, 2018.

[8] Xiaofei Wang, Ruizhi Li, and Hynek Hermansky, "Stream attention for distributed multi-microphone speech recognition," in *Proc. of INTERSPEECH*, 2018, pp. 3033–3037.

[9] Feifei Xiong et al., "Channel selection using neural network posterior probability for speech recognition with distributed microphone arrays in everyday environments," in *CHiME-5*, 2018.

[10] Jonathan G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc. of ASRU*. IEEE, 1997, pp. 347–354.

[11] Takuya Yoshioka et al., "Meeting transcription using asynchronous distant microphones," in *Proc. Interspeech 2019*, 2019, pp. 2968–2972.

[12] Jun Du et al., "The ustc-iflytek systems for chime-5 challenge," in *CHiME-5*, 2018.

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, 2006, pp. 369–376.

[14] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. of ICML*, 2014, pp. 1764–1772.

[15] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. of ASRU*, 2015.

[16] William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. of ICASSP*, 2015.

[17] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. of NIPS*, 2015, pp. 577–585.

[18] Alex Graves, "Sequence transduction with recurrent neural networks," in *Proc. of ICML Workshop on Representation Learning*, 2012.

[19] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of ICASSP*. IEEE, 2013, pp. 6645–6649.

[20] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. of ICASSP*, 2017, pp. 4835–4839.

[21] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM," in *Proc. of INTERSPEECH*, 2017.

[22] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[23] Ruizhi Li, Xiaofei Wang, Sri Harish Mallidi, Shinji Watanabe, Takaaki Hori, and Hynek Hermansky, "Multi-stream end-to-end speech recognition," *arXiv preprint arXiv:1906.08041*, 2019.

[24] Ruizhi Li et al., "Multi-encoder multi-resolution framework for end-to-end speech recognition," *arXiv preprint arXiv:1811.04897*, 2018.

[25] Xiaofei Wang, Ruizhi Li, Sri Harish Mallidi, Takaaki Hori, Shinji Watanabe, and Hynek Hermansky, "Stream attention-based multi-array end-to-end speech recognition," in *Proc. of ICASSP*. IEEE, 2019, pp. 7105–7109.

[26] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. of NIPS*, pp. 577–585. Curran Associates, Inc., 2015.

[27] Mirco Ravanelli, Piergiorgio Svaizer, and Maurizio Omologo, "Realistic multi-microphone data simulation for distant speech recognition," in *Proc. of INTERSPEECH*, 2016.

[28] Jean Carletta et al., "The ami meeting corpus: A pre-announcement," in *Proc. of MLMI*. Springer, 2005, pp. 28–39.

[29] Xavier Anguera, Chuck Wooters, and Javier Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[30] Shinji Watanabe et al., "Espnet: End-to-end speech processing toolkit," in *Proc. of INTERSPEECH*, 2018, pp. 2207–2211.

[31] Jaejin Cho et al., "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *Proc. of SLT*, 2018.

[32] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017.

[33] Takaaki Hori, Jaejin Cho, and Shinji Watanabe, "End-to-end speech recognition with word-based RNN language models," *arXiv preprint arXiv:1808.02608*, 2018.

[34] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. of INTERSPEECH*, 2019.