

END-TO-END NEURAL SPEAKER DIARIZATION WITH SELF-ATTENTION

Yusuke Fujita^{1,2}, Naoyuki Kanda¹, Shota Horiguchi¹, Yawen Xue¹, Kenji Nagamatsu¹, Shinji Watanabe²

¹ Hitachi, Ltd. Research & Development Group, Japan

² Center for Language and Speech Processing, Johns Hopkins University, USA

ABSTRACT

Speaker diarization has been mainly developed based on the clustering of speaker embeddings. However, the clustering-based approach has two major problems; i.e., (i) it is not optimized to minimize diarization errors directly, and (ii) it cannot handle speaker overlaps correctly. To solve these problems, the End-to-End Neural Diarization (EEND), in which a bidirectional long short-term memory (BLSTM) network directly outputs speaker diarization results given a multi-talker recording, was recently proposed. In this study, we enhance EEND by introducing self-attention blocks instead of BLSTM blocks. In contrast to BLSTM, which is conditioned only on its previous and next hidden states, self-attention is directly conditioned on all the other frames, making it much suitable for dealing with the speaker diarization problem. We evaluated our proposed method on simulated mixtures, real telephone calls, and real dialogue recordings. The experimental results revealed that the self-attention was the key to achieving good performance and that our proposed method performed significantly better than the conventional BLSTM-based method. Our method was even better than that of the state-of-the-art x-vector clustering-based method. Finally, by visualizing the latent representation, we show that the self-attention can capture global speaker characteristics in addition to local speech activity dynamics. Our source code is available online at <https://github.com/hitachi-speech/EEND>.

Index Terms— speaker diarization, neural network, end-to-end, self-attention

1. INTRODUCTION

Speaker diarization is the process of partitioning an audio recording into homogeneous segments according to the speaker's identity. The speaker diarization has a wide range of applications, such as information retrieval from broadcast news, generating minutes of meetings, and a turn-taking analysis of telephone conversations [1, 2]. It also helps automatic speech recognition performance in multi-speaker conversation scenarios in meetings (ICSI [3, 4], AMI [5, 6]) and home environments (CHiME-5 [6–10]).

Typical speaker diarization systems are based on the clustering of speaker embeddings [11–18]. For instance, i-vectors [12, 13, 17, 19], d-vectors [18, 20], and x-vectors [16, 21] are commonly used in speaker diarization tasks. These embeddings of short segments are partitioned into speaker clusters by using clustering algorithms, such as Gaussian mixture models [11, 12], agglomerative hierarchical clustering [11, 13, 16, 17], mean shift clustering [14], k-means clustering [15, 18], Links [18, 22], and spectral clustering [18]. These clustering-based diarization methods have shown themselves to be

effective on various datasets (see the DIHARD Challenge 2018 activities, e.g., [23–25]).

However, such clustering-based methods have a number of problems. First, they cannot be optimized to minimize diarization errors directly, because the clustering procedure is a type of unsupervised learning methods. Second, they have trouble handling speaker overlaps, since the clustering algorithms implicitly assume one speaker per segment. Furthermore, they have trouble adapting their speaker embedding models to real audio recordings with speaker overlaps, because the speaker embedding model has to be optimized with single-speaker non-overlapping segments. These problems hinder the speaker diarization application from working on real audio recordings that usually contain overlapping segments.

To solve these problems, we propose Self-Attentive End-to-End Neural Diarization (SA-EEND). Different from most of the other methods, our proposed method does not rely on clustering. Instead, a self-attention-based neural network directly outputs the joint speech activities of all speakers for each time frame, given an input of a multi-speaker audio recording. Our method can naturally handle speaker overlaps during the training and inference time by exploiting a multi-label classification framework. The neural network is trained in an end-to-end fashion using a recently proposed permutation-free objective function that provides minimal diarization errors [26].

This paper shows that our method achieves a significant performance improvement over end-to-end neural diarization (EEND) [26], for which promising but preliminary results were reported with a bidirectional long short-term memory (BLSTM) [27]. In particular, it shows that the self-attention mechanism [28, 29] is the key to achieving good speaker-diarization performance in this paper. We demonstrate that the self-attention mechanism gives significantly better results for multiple datasets compared with the BLSTM-based method [26] and the state-of-the-art x-vector-based speaker diarization method. In contrast to BLSTM, which is conditioned only on its previous and next hidden states, the self-attention layer is conditioned on all the other input frames by computing the pairwise similarity between all frame pairs. We believe that this mechanism is the key to speaker diarization since it can capture global speaker characteristics in addition to local speech activity dynamics. By visualizing the learned representation, we show that some self-attention heads capture speaker-dependent global characteristics, while the remaining heads represent temporal features.

2. RELATED WORK

2.1. Clustering-based methods

The x-vector clustering-based system is commonly used for speaker diarization [23, 24, 30]. A diagram of the system is depicted in Fig. 1(a). To build the system, one has to prepare three independent models: (i) a speech activity detection (SAD) neural network, (ii)

The first author performed the work while at Center for Language and Speech Processing, Johns Hopkins University as a Visiting Scholar.

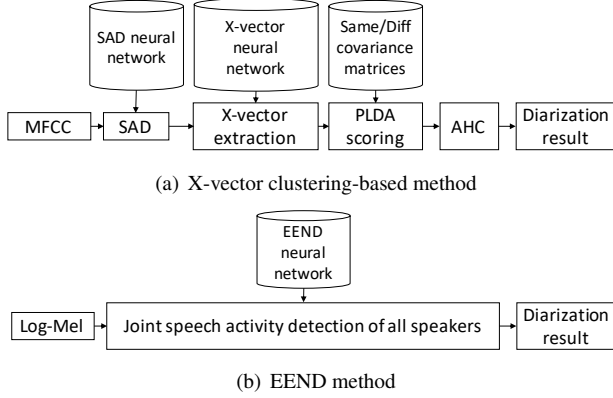


Fig. 1. System diagrams for speaker diarization

x-vector extraction neural network, and (iii) PLDA model including the same/different speaker covariance matrices. None of these models can be trained to directly minimize the diarization errors. Joint modeling methods have been studied in an effort to alleviate the complex preparation process and take into account the dependencies between these models. They include, for example, joint modeling of x-vector extraction and PLDA scoring [16, 31] and joint modeling of SAD and speaker embedding [32]. However, the clustering process has remained unchanged because it is an unsupervised process.

In contrast to these methods, the EEND method uses only one neural network model, as depicted in Fig. 1(b). This method does not rely on clustering, and the model can be directly optimized with the reference diarization results of the training data.

This neural-network-based end-to-end approach, in which only one neural network model directly computes the final outputs, has been successfully applied in a variety of tasks, including neural machine translation [33, 34], automatic speech recognition [35–37], and text-to-speech [38, 39].

2.2. Direct optimization minimizing diarization errors

A fully supervised diarization method has been proposed for optimization based on a diarization error minimization objective [40]. This is the first successful approach that does not cluster speaker embeddings. The method formulates the speaker diarization problem on the basis of a factored probabilistic model, which consists of modules for determining speaker changes, speaker assignments, and feature generation. These models are jointly trained using input features and corresponding speaker labels. However, the SAD model and their speaker embedding (d-vector) model have to be trained separately in their method. Moreover, their speaker-change model assumes one speaker for each segment, which hinders its application to speaker-overlapping speech.

In contrast to their method, the EEND method uses an end-to-end neural network that accepts audio features as input and outputs the joint speech activities of multiple speakers. The network is optimized using the entire recording, including non-speech and speaker overlaps, with a diarization-error-oriented objective. This end-to-end model was first introduced in [26]; this paper describes an extension of the model that includes a self-attention mechanism.

2.3. Self-attention mechanism

The self-attention mechanism was originally proposed for extracting sentence embeddings for text processing [28]. Recently, the self-attention mechanism has shown superior performance in a variety of tasks, including machine translation [29], video classification [41], and image segmentation [42]. For audio processing, a self-attention mechanism has been incorporated in acoustic modeling for ASR [43, 44], sound event detection [45], and speaker recognition [46]. For speaker diarization, the self-attention mechanism has been applied to the speaker embedding extraction model [25] and the scoring model [31] of clustering-based methods. This study describes a self-attention mechanism for clustering-free speaker diarization.

3. PROPOSED METHOD: SELF-ATTENTIVE END-TO-END NEURAL DIARIZATION

3.1. End-to-end neural diarization: review

Here, we describe the EEND method proposed in [26]. The speaker diarization task can be formulated as a multi-label classification problem, as follows.

Given a T -length observation sequence $X = (\mathbf{x}_t \in \mathbb{R}^F \mid t = 1, \dots, T)$ from an audio signal, speaker diarization problem tries to estimate the corresponding speaker label sequence $Y = (\mathbf{y}_t \mid t = 1, \dots, T)$. Here, \mathbf{x}_t is a F -dimensional observation feature vector at time index t . Speaker label $\mathbf{y}_t = [y_{t,c} \in \{0, 1\} \mid c = 1, \dots, C]$ denotes a joint activity for multiple (C) speakers at time index t . For example, $y_{t,c} = 1$ and $y_{t,c'} = 1$ ($c \neq c'$) represent an overlap situation in which speakers c and c' are both present at time index t . Thus, determining Y is a sufficient condition to determine the speaker diarization information.

The most probable speaker label sequence \hat{Y} is selected from among all possible speaker label sequences \mathcal{Y} , as follows:

$$\hat{Y} = \arg \max_{Y \in \mathcal{Y}} P(Y|X). \quad (1)$$

$P(Y|X)$ can be factorized using the conditional independence assumption as follows:

$$P(Y|X) = \prod_t P(\mathbf{y}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}, X), \quad (2)$$

$$\approx \prod_t P(\mathbf{y}_t | X) \approx \prod_t \prod_c P(y_{t,c} | X). \quad (3)$$

Here, we assume that the frame-wise posterior is conditioned on all inputs, and each speaker is present independently. The frame-wise posterior $P(y_{t,c} | X)$ can be estimated using a neural-network-based model.

3.2. Self-attention-based neural network

In [26], a BLSTM based neural network was used for estimating the frame-wise posteriors $P(y_{t,c} | X)$. In this paper, we propose self-attentive end-to-end neural diarization (SA-EEND), which uses self-attention-based encoding blocks instead of BLSTMs, as depicted in Fig. 2. The input features are transformed as follows:

$$\mathbf{e}_t^{(0)} = \mathbf{W}_0 \mathbf{x}_t + \mathbf{b}_0 \in \mathbb{R}^D, \quad (4)$$

$$\mathbf{e}_t^{(p)} = \text{Encoder}_t^{(p)}(\mathbf{e}_1^{(p-1)}, \dots, \mathbf{e}_T^{(p-1)}) \quad (1 \leq p \leq P). \quad (5)$$

Here, $\mathbf{W}_0 \in \mathbb{R}^{D \times F}$ and $\mathbf{b}_0 \in \mathbb{R}^D$ project an input feature into D -dimensional vector. $\text{Encoder}_t^{(p)}(\cdot)$ is the p -th encoder block which accepts an input sequence of D -dimensional vectors and outputs a D -dimensional vector $\mathbf{e}_t^{(p)}$ at time index t . We use P encoder blocks followed by the output layer for frame-wise posteriors.

The architecture of the encoder block is depicted in Fig. 2. This configuration of the encoder block is almost the same as the one in the Speech-Transformer introduced in [44], but without positional encoding. The encoder block has two sub-layers. The first is a multi-head self-attention layer, and the second is a position-wise feed-forward layer.

3.2.1. Multi-head self-attention layer

The multi-head self-attention layer transforms a sequence of input vectors as follows. The sequence of vectors $(\mathbf{e}_t^{(p-1)})_{t=1, \dots, T}$ is converted into a $\mathbb{R}^{T \times D}$ matrix, followed by layer normalization [47]:

$$\bar{\mathbf{E}}^{(p-1)} = \text{LayerNorm}([\mathbf{e}_1^{(p-1)} \dots \mathbf{e}_T^{(p-1)}]^\top) \in \mathbb{R}^{T \times D}. \quad (6)$$

Then, for each head, a pairwise similarity matrix $\mathbf{A}_h^{(p)}$ is computed using the dot products of query vectors $\bar{\mathbf{E}}^{(p-1)} \mathbf{Q}_h^{(p)} \in \mathbb{R}^{T \times d}$ and key vectors $\bar{\mathbf{E}}^{(p-1)} \mathbf{K}_h^{(p)} \in \mathbb{R}^{T \times d}$:

$$\mathbf{A}_h^{(p)} = \bar{\mathbf{E}}^{(p-1)} \mathbf{Q}_h^{(p)} (\bar{\mathbf{E}}^{(p-1)} \mathbf{K}_h^{(p)})^\top \in \mathbb{R}^{T \times T} \quad (1 \leq h \leq H), \quad (7)$$

where, $\mathbf{Q}_h^{(p)}, \mathbf{K}_h^{(p)} \in \mathbb{R}^{D \times d}$ are query and key projection matrices for the h -th head, respectively. $d = D/H$ is a dimension of each head, and H is the number of heads. The pairwise similarity matrix $\mathbf{A}_h^{(p)}$ is scaled by $1/\sqrt{d}$ and a softmax function is applied to form the attention weight matrix $\hat{\mathbf{A}}_h^{(p)}$:

$$\hat{\mathbf{A}}_h^{(p)} = \text{Softmax} \left(\frac{\mathbf{A}_h^{(p)}}{\sqrt{d}} \right) \in \mathbb{R}^{T \times T}. \quad (8)$$

Then, using the attention weight matrix, context vectors $\mathbf{C}_h^{(p)}$ are computed as a weighted sum of the value vectors $\bar{\mathbf{E}}^{(p-1)} \mathbf{V}_h^{(p)} \in \mathbb{R}^{T \times d}$:

$$\mathbf{C}_h^{(p)} = \hat{\mathbf{A}}_h^{(p)} (\bar{\mathbf{E}}^{(p-1)} \mathbf{V}_h^{(p)}) \in \mathbb{R}^{T \times d}, \quad (9)$$

where $\mathbf{V}_h \in \mathbb{R}^{D \times d}$ is the value projection matrix. Finally, the context vectors for all heads are concatenated and projected using the output projection matrix $\mathbf{O}^{(p)} \in \mathbb{R}^{D \times D}$:

$$\mathbf{E}^{(p,SA)} = [\mathbf{C}_1^{(p)} \dots \mathbf{C}_H^{(p)}] \mathbf{O}^{(p)} \in \mathbb{R}^{T \times D}. \quad (10)$$

Following the self-attention layer, a residual connection and layer normalization is applied:

$$\bar{\mathbf{E}}^{(p,SA)} = \text{LayerNorm}(\bar{\mathbf{E}}^{(p-1)} + \mathbf{E}^{(p,SA)}) \in \mathbb{R}^{T \times D}. \quad (11)$$

3.2.2. Position-wise feed-forward layer

The position-wise feed-forward layer transforms $\bar{\mathbf{E}}^{(p,SA)}$ as follows:

$$\mathbf{E}^{(p,FF)} = \text{ReLU}(\bar{\mathbf{E}}^{(p,SA)} \mathbf{W}_1^{(p)} + \mathbf{b}_1^{(p)} \mathbf{1}) \mathbf{W}_2^{(p)} + \mathbf{b}_2^{(p)} \mathbf{1} \in \mathbb{R}^{T \times D}, \quad (12)$$

where $\mathbf{W}_1^{(p)} \in \mathbb{R}^{D \times d_{\text{ff}}}$ and $\mathbf{b}_1^{(p)} \in \mathbb{R}^{d_{\text{ff}}}$ are the first linear projection matrix and bias, respectively, $\mathbf{1} \in \mathbb{R}^{1 \times T}$ is an all-one row vector, and $\text{ReLU}(\cdot)$ is the rectified linear unit activation function.

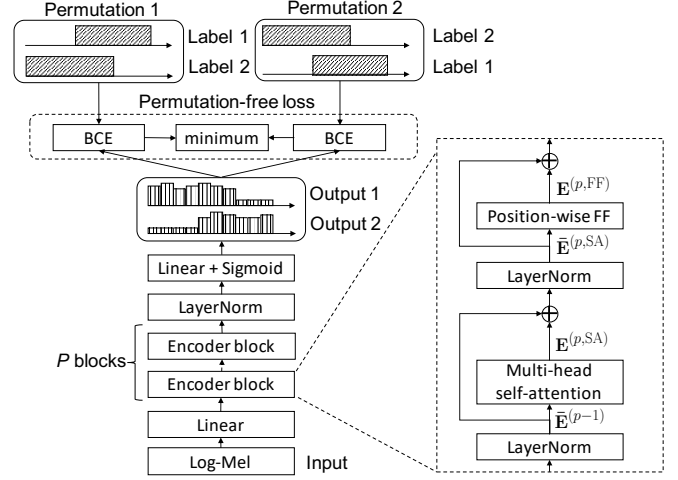


Fig. 2. Two-speaker SA-EEND model trained with permutation-free loss.

d_{ff} is the number of internal units in this layer. $\mathbf{W}_2^{(p)} \in \mathbb{R}^{d_{\text{ff}} \times D}$ and $\mathbf{b}_2^{(p)} \in \mathbb{R}^D$ are the second linear projection matrix and bias, respectively.

Finally, the output of the encoder block $\mathbf{e}_t^{(p)}$ for each time frame is computed by applying a residual connection as follows:

$$[\mathbf{e}_1^{(p)} \dots \mathbf{e}_T^{(p)}] = (\bar{\mathbf{E}}^{(p,SA)} + \mathbf{E}^{(p,FF)})^\top \quad (13)$$

3.2.3. Output layer for frame-wise posteriors

The frame-wise posteriors \mathbf{z}_t are calculated from $\mathbf{e}_t^{(P)}$ (in Eq. 5) using layer normalization and a fully-connected layer as follows:

$$\bar{\mathbf{E}}^{(P)} = \text{LayerNorm}([\mathbf{e}_1^{(P)} \dots \mathbf{e}_T^{(P)}]^\top) \in \mathbb{R}^{T \times D}, \quad (14)$$

$$[\mathbf{z}_1 \dots \mathbf{z}_T] = \sigma(\bar{\mathbf{E}}^{(P)} \mathbf{W}_3 + \mathbf{b}_3 \mathbf{1})^\top, \quad (15)$$

where $\mathbf{W}_3 \in \mathbb{R}^{D \times C}$ and $\mathbf{b}_3 \in \mathbb{R}^C$ are the linear projection matrix and bias, respectively, and $\sigma(\cdot)$ is the element-wise sigmoid function.

3.3. Permutation-free training

The difficulty of training the model described above is that the model must deal with speaker permutations: changing the order of speakers within a correct label sequence is also regarded as correct. An example of permutations in a two-speaker case is shown in Fig. 2. In this paper, we call this problem “label ambiguity.” This label ambiguity obstructs the training of the neural network when we use a standard binary cross entropy loss function.

To cope with the label ambiguity problem, the permutation-free training scheme considers all the permutations of the reference speaker labels. The permutation-free training scheme has been used in research on source separation [48–50]. Here, we apply the permutation-free loss function to a temporal sequence of speaker labels. The neural network is trained to minimize the permutation-free loss between the output \mathbf{z}_t predicted in Eq. 15 and the reference speaker label \mathbf{l}_t , as follows:

$$J^{\text{PF}} = \frac{1}{TC} \min_{\phi \in \text{perm}(C)} \sum_t \text{BCE}(\mathbf{l}_t^\phi, \mathbf{z}_t), \quad (16)$$

Table 1. Statistics of training and test sets.

	# mixtures	avg. duration (sec)	overlap ratio (%)
Traning sets			
Simulated ($\beta = 2$)	100,000	87.6	34.4
Real (SWBD+SRE)	26,172	304.7	3.7
Test sets			
Simulated ($\beta = 2$)	500	87.3	34.4
Simulated ($\beta = 3$)	500	103.8	27.2
Simulated ($\beta = 5$)	500	137.1	19.5
CALLHOME [51]	148	72.1	13.0
CSJ [52]	54	766.3	20.1

where $\text{perm}(C)$ is the set of all the possible permutations of $(1, \dots, C)$, and \mathbf{l}_t^ϕ is the ϕ -th permutation of the reference speaker label, and $\text{BCE}(\cdot, \cdot)$ is the binary cross entropy function between the label and the output.

4. EXPERIMENTAL SETUP

4.1. Data

To verify the effectiveness of the SA-EEND method for various overlap situations, we prepared two training sets and five test sets, including simulated and real datasets. The statistics of the training and test sets are listed in Table 1. The overlap ratio is computed as the ratio of the audio time during which two or more speakers are active, to the audio time during which one or more speakers are active.

Note that training data for the EEND method is different from those for the x-vector clustering-based method. Whereas the x-vector clustering-based method uses single-speaker segments for training their x-vector neural network, the EEND method uses audio mixtures of multiple speakers. Such mixtures can be simulated infinitely with a combination of single-speaker segments. Moreover, the EEND model can be trained with not only simulated mixtures but real audio mixtures with speaker overlaps.

4.1.1. Simulated mixtures

Each mixture was simulated by Algorithm 1. Unlike the mixture simulation of source separation studies [48], we consider a diarization-style mixture: each speech mixture should have dozens of utterances per speaker with reasonable silence intervals between utterances. The silence intervals are controlled by the average interval of β . Larger values of β generate speech with less overlap.

The set of utterances used for the simulation was comprised of the Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, Part2), and NIST Speaker Recognition Evaluation datasets (2004, 2005, 2006, 2008). All recordings are telephone speech sampled at 8 kHz. There are 6,381 speakers in total. We split them into 5,743 speakers for the training set and 638 speakers for the test set. Note that the set of utterances for the training set is identical to that of the Kaldi CALLHOME diarization v2 recipe [53]¹, making it fair comparison with the x-vector clustering-based method.

Since there are no time annotations in these corpora, we extracted utterances using speech activity detection (SAD) on the basis

¹https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization

Algorithm 1: Mixture simulation.

Input: $\mathcal{S}, \mathcal{N}, \mathcal{I}, \mathcal{R}$ // Set of speakers, noises, RIRs and SNRs
 $\mathcal{U} = \{U_s\}_{s \in \mathcal{S}}$ // Set of utterance lists
 N_{spk} // #speakers per mixture
 $N_{\text{umax}}, N_{\text{umin}}$ // Max. and min. #utterances per speaker
 β // Average interval
// Mixture

Output: \mathbf{y}

- 1 Sample a set of N_{spk} speakers \mathcal{S}' from \mathcal{S}
- 2 $\mathcal{X} \leftarrow \emptyset$ // Set of N_{spk} speakers' signals
- 3 **forall** $s \in \mathcal{S}'$ **do**
- 4 $\mathbf{x}_s \leftarrow \emptyset$ // Concatenated signal
- 5 Sample \mathbf{i} from \mathcal{I} // RIR
- 6 Sample N_u from $\{N_{\text{umin}}, \dots, N_{\text{umax}}\}$
- 7 **for** $u = 1$ to N_u **do**
- 8 Sample $\delta \sim \frac{1}{\beta} \exp\left(-\frac{\delta}{\beta}\right)$ // Interval
- 9 $\mathbf{x}_s \leftarrow \mathbf{x}_s \oplus \mathbf{0}^{(\delta)} \oplus U_s[u] * \mathbf{i}$
- 10 $\mathcal{X}.\text{add}(\mathbf{x}_s)$
- 11 $L_{\text{max}} = \max_{\mathbf{x} \in \mathcal{X}} |\mathbf{x}|$
- 12 $\mathbf{y} \leftarrow \sum_{\mathbf{x} \in \mathcal{X}} (\mathbf{x} \oplus \mathbf{0}^{(L_{\text{max}} - |\mathbf{x}|)})$
- 13 Sample \mathbf{n} from \mathcal{N} // Background noise
- 14 Sample r from \mathcal{R} // SNR
- 15 Determine a mixing scale p from r, \mathbf{y} , and \mathbf{n}
- 16 $\mathbf{n}' \leftarrow$ repeat \mathbf{n} until reach the length of \mathbf{y}
- 17 $\mathbf{y} \leftarrow \mathbf{y} + p \cdot \mathbf{n}'$

of time-delay neural networks and statistics pooling².

The set of background noises was from the MUSAN corpus [54]. We used 37 recordings that are annotated as “background” noises. The set of 10,000 room impulse responses (RIRs) was from the Simulated Room Impulse Response Database used in [55]. The SNR values were sampled from 10, 15, and 20 dBs. These sets of non-speech corpora are also used for training the x-vector and SAD models in the x-vector clustering-based method.

We generated two-speaker mixtures for each speaker with 10-20 utterances ($N_{\text{spk}} = 2, N_{\text{umin}} = 10, N_{\text{umax}} = 20$). For the simulated training set, 100,000 mixtures were generated with $\beta = 2$. For the simulated test set, 500 mixtures were generated with $\beta = 2, 3$, and 5. The overlap ratios of the simulated mixtures are ranging from 19.5 to 34.4%.

4.1.2. Real datasets

We used real telephone speech recordings as the real training set. A set of 26,172 two-speaker recordings were extracted from the recordings of the Switchboard-2 (Phase I, II, III), Switchboard Cellular (Part 1, Part 2), and NIST Speaker Recognition Evaluation datasets. The overlap ratio of the training data was 3.7%, far less than that of the simulated mixtures.

We evaluated the proposed method on real telephone conversations in the CALLHOME dataset [51]. We randomly split the two-speaker recordings from the CALLHOME dataset into two subsets: an adaptation set of 155 recordings and a test set of 148 recordings. The average overlap ratio of the test set was 13.0%.

In addition, we conducted an evaluation on the dialogue part of the Corpus of Spontaneous Japanese (CSJ) [52]. The CSJ con-

²The SAD model: <http://kaldi-asr.org/models/m4>

tains 54 two-speaker dialogue recordings³. They were recorded using headset microphones in separate soundproof rooms. The average overlap ratio of the CSJ test set was 20.1%, larger than the CALLHOME test set.

4.2. Model configuration

4.2.1. Clustering-based systems

We compared the proposed method with two conventional clustering-based systems [23]: the i-vector system and x-vector system were created using the Kaldi CALLHOME diarization v1 and v2 recipes.

These recipes use agglomerative hierarchical clustering (AHC) with the probabilistic linear discriminant analysis (PLDA) scoring scheme. The number of clusters was fixed to 2. Though the original recipes use oracle speech/non-speech marks, we used the SAD model with the same configuration as described in Sec. 4.1.

4.2.2. BLSTM-based EEND system

We configured a BLSTM-based EEND method (BLSTM-EEND), as described in [26]. The input features were 23-dimensional log-Mel-filterbanks with a 25-ms frame length and 10-ms frame shift. Each feature was concatenated with those from the previous seven frames and subsequent seven frames. To deal with a long audio sequence in our neural networks, we subsampled the concatenated features by a factor of ten. Consequently, a (23×15) -dimensional input feature was fed into the neural network every 100 ms.

We used a five-layer BLSTM with 256 hidden units in each layer. The second layer of the BLSTM outputs was used to form a 256-dimensional embedding; we then calculated the deep clustering loss in this embedding to discriminate different speakers. We used the Adam [56] optimizer with a learning rate of 10^{-3} . The batch size was 10. The number of training epochs was 20.

Because the output of the neural network is the probability of speech activity for each speaker, a threshold is required to obtain the decision of speech activity for each frame. We set the threshold to 0.5. Furthermore, we applied 11-frame median filtering to prevent production of unreasonably short segments.

For domain adaptation, the neural network was retrained using the CALLHOME adaptation set. we used the Adam optimizer with a learning rate of 10^{-6} and ran 5 epochs. For the postprocessing, we adjusted the threshold to 0.6 so that the DER of the adaptation set has the minimum value.

4.2.3. Self-attentive EEND system

Here, we used the same input features as were input to the BLSTM-EEND system. Note that the sequence length at the training stage was limited to 500 (50 seconds in audio time) because our system uses more memory than the BLSTM-based network does. Therefore, we split the input audio recordings into non-overlapping 50-second segments. At the inference stage, we used the entire sequence for each recording.

We used two encoder blocks with 256 attention units containing four heads ($P = 2$, $D = 256$, $H = 4$). We used 1024 internal units in a position-wise feed-forward layer ($d_{ff} = 1024$). We used the Adam optimizer with the learning rate scheduler introduced in [29]. The number of warm-up steps used in the learning rate scheduler was 25,000. The batch size was 64. The number of training epochs

³We excluded four out of 58 recordings that contain speakers in the official speech recognition evaluation sets.

Table 2. DERs (%) on various test sets. For EEND systems, the CALLHOME (CH) results are obtained with domain adaptation.

	Simulated			Real	
	$\beta = 2$	$\beta = 3$	$\beta = 5$	CH	CSJ
Clustering-based					
i-vector	33.74	30.93	25.96	12.10	27.99
x-vector	28.77	24.46	19.78	11.53	22.96
BLSTM-EEND					
trained with sim.	12.28	14.36	19.69	26.03	39.33
trained with real	36.23	37.78	40.34	23.07	25.37
SA-EEND					
trained with sim.	7.91	8.51	9.51	13.66	22.31
trained with real	32.72	33.84	36.78	10.76	20.50

Table 3. DERs (%) on the CALLHOME with and without domain adaptation.

	w/o adaptation	with adaptation
x-vector clustering	11.53	N/A
BLSTM-EEND		
trained with sim.	43.84	26.03
trained with real	31.01	23.07
SA-EEND		
trained with sim.	17.42	13.66
trained with real	12.66	10.76

was 100. After 100 epochs, we used an averaged model obtained by averaging the model parameters of the last 10 epochs. As with the BLSTM-EEND system, we applied 11-frame median filtering.

For domain adaptation, the averaged model was retrained using the CALLHOME adaptation set. We used the Adam optimizer with a learning rate of 10^{-5} and ran 100 epochs. After 100 epochs, we used an averaged model obtained by averaging the model parameters of the last 10 epochs.

4.3. Performance metric

We evaluated the systems with the diarization error rate (DER) [57]. Note that the DERs reported in many prior studies did not include misses or false alarm errors due to their using oracle speech/non-speech labels. Overlapping speech segments had also been excluded from the evaluation. For our DER computation, we evaluated all of the errors, including overlapping speech segments, because the proposed method includes both the speech activity detection and overlapping speech detection functionality. As is done typically, we used a collar tolerance of 250 ms at the start and end of each segment.

5. RESULTS

5.1. Evaluation on simulated mixtures

DERs on various test sets are shown in Table 2. The clustering-based systems performed poorly on heavily overlapped simulated mixtures. This result is within our expectations, because the clustering-based systems did not consider speaker overlaps; there are more misses when the overlap ratio is high.

The BLSTM-EEND system trained with the simulated training set showed a significant DER reduction compared with the clustering-based systems on the simulated mixtures. Among the differing overlap ratios, it showed the best performance on the highest

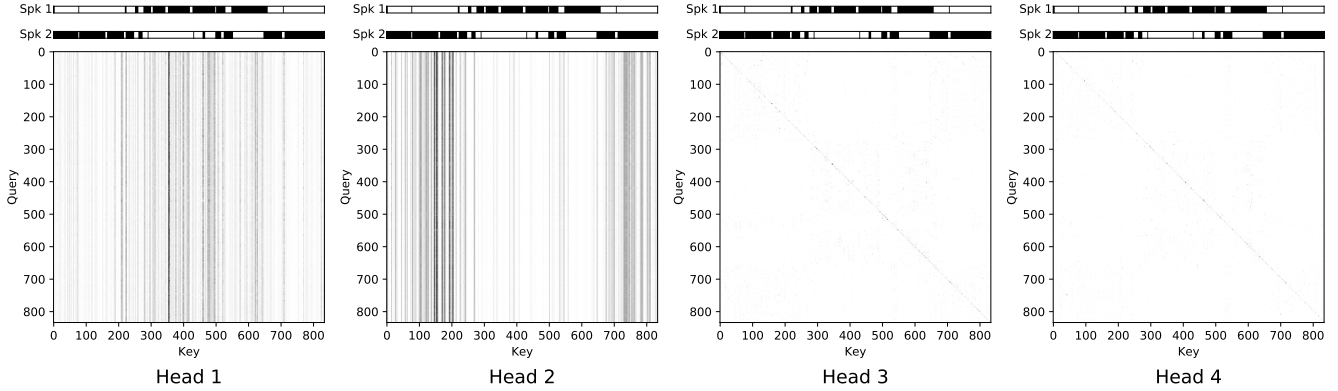


Fig. 3. Attention weight matrices at the second encoder block. The input was the CALLHOME test set (recording id: iagk). The model was trained with the real training set followed by domain adaptation. The top two rows show the reference speech activity of two speakers.

Table 4. Detailed DERs (%) evaluated on the CALLHOME. DER is composed of Misses (MI), False alarms (FA), and Confusion errors (CF). The SAD errors are composed of Misses (MI) and False alarms (FA) errors.

Method	DER	DER breakdown			SAD errors	
		MI	FA	CF	MI	FA
i-vector	12.10	7.74	0.54	3.82	1.4	0.5
x-vector	11.53	7.74	0.54	3.25	1.4	0.5
SA-EEND						
no-adapt	12.66	7.42	3.93	1.31	3.3	0.6
adapted	10.76	6.68	2.40	1.68	2.3	0.5

overlap ratio condition ($\beta = 2$). The BLSTM-EEND system worked well on the overlapping condition matched with training data.

The proposed system, SA-EEND, trained with the simulated training set had significantly fewer DERs compared with the BLSTM-EEND system on every test set. As well as the BLSTM-EEND system, it showed the best performance on the highest overlap ratio condition ($\beta = 2$). However, the DER degradation on the less overlapping conditions was smaller than that of the BLSTM-EEND system, which indicated that the self-attention blocks improved robustness to variable overlapping conditions.

5.2. Evaluation on real test sets

In contrast to the good performance on the simulated mixtures, the BLSTM-EEND system had inferior DERs to those of the clustering-based systems evaluated on the real test sets. Although the BLSTM-EEND system showed performance improvements when the training data were switched from simulated to real data, its DERs were still higher than those of the clustering-based systems.

The proposed system, SA-EEND, trained with the simulated training set showed remarkable improvements on real datasets of the CALLHOME and CSJ, which indicates the strong generalization capability of the self-attention blocks. For the CSJ, even without domain adaptation, the proposed system performed better than the x-vector clustering-based method.

The SA-EEND system trained with the real training set performed the best on the real test sets, however, it had poor DERs on

the simulated mixtures. We expected that the result was due to the small number of mixtures and low overlap ratio of the real training set. It would be much improved by feeding more real data with more speaker overlaps, or by combining with simulated training data.

5.3. Effect of domain adaptation

The EEND models trained with simulated training set were overfitted to the specific overlap ratio of the training set. We expected that the overfitting would be mitigated by using domain adaptation. DERs on the CALLHOME with and without domain adaptation are shown in Table 3. As expected, the domain adaptation significantly reduced the DER; our system thus achieved even better results than those of the x-vector-based system.

A detailed DER comparison on the CALLHOME test set is shown in Table 4. The clustering-based systems had few SAD errors thanks to the robust SAD model trained with various noise-augmented data. However, there were numerous misses and confusion errors due to its lack of handling speaker overlaps. Compared with clustering-based systems, the proposed method produced significantly fewer confusion and miss errors. The domain adaptation reduced all error types except confusion errors.

5.4. Visualization of self-attention

To analyze the behavior of the self-attention mechanism in our diarization system, Fig. 3 visualizes the attention weight matrix at the second encoder block, corresponding to $\hat{\mathbf{A}}_h^{(p=2)}$ in Eq. 8. Here, head 1 and head 2 have vertical lines at different positions. The vertical lines correspond to each speaker’s activity. The attention weight matrix with these vertical lines transformed the input features into the weighted mean of the same speaker frames. These heads actually captured the global speaker characteristics by computing the similarity between distant frames. Interestingly, heads 3 and 4 look like identity matrices, which results in position-independent linear transforms. These heads are considered to work for speech/non-speech detection. We conclude that the multi-head self-attention mechanism captures global speaker characteristics in addition to local speech activity dynamics, which leads to a reduction in DER. Experiments on various combinations of the number of heads and the number of speakers would be an interesting future work.

6. CONCLUSION

We incorporated a self-attention mechanism in the end-to-end neural diarization model. We evaluated our model on simulated mixtures and two real datasets. Experimental results showed that the self-attention mechanism significantly reduced DERs and showed higher generalization quality compared with a BLSTM-based neural diarization system. The self-attention based systems even outperformed x-vector clustering-based systems. We also showed that the self-attention blocks actually captured global speaker characteristics by visualizing the latent representation.

7. REFERENCES

- [1] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. on ASLP*, vol. 14, no. 5, pp. 1557–1565, 2006.
- [2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. on ASLP*, vol. 20, no. 2, pp. 356–370, 2012.
- [3] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proc. ICASSP*, 2003, vol. I, pp. 364–367.
- [4] Ö. Çetin and E. Shriberg, "Overlap in meetings: ASR effects and analysis by dialog factors, speakers, and collection site," in *Proc. MLMI*, 2006, pp. 212–224.
- [5] S. Renals, T. Hain, and H. Bourlard, "Interpretation of multi-party meetings the AMI and Amida projects," in *2008 Hands-Free Speech Communication and Microphone Arrays*, 2008, pp. 115–118.
- [6] N. Kanda, Y. Fujita, S. Horiguchi, R. Ikeshita, K. Nagamatsu, and S. Watanabe, "Acoustic modeling for distant multi-talker speech recognition with single- and multi-channel branches," in *Proc. ICASSP*, 2019, pp. 6630–6634.
- [7] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.
- [8] J. Du, T. Gao, L. Sun, F. Ma, Y. Fang, D.-Y. Liu, Q. Zhang, X. Zhang, H.-K. Wang, J. Pan, J.-Q. Gao, C.-H. Lee, and J.-D. Chen, "The USTC-iFlytek Systems for CHiME-5 Challenge," in *Proc. CHiME-5*, 2018, pp. 11–15.
- [9] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-End Processing for the CHiME-5 Dinner Party Scenario," in *Proc. CHiME-5*, 2018, pp. 35–40.
- [10] N. Kanda, R. Ikeshita, S. Horiguchi, Y. Fujita, K. Nagamatsu, X. Wang, V. Manohar, N. E. Yalta Soplin, M. Maciejewski, S.-J. Chen, A. S. Subramanian, R. Li, Z. Wang, J. Naradowsky, L. P. Garcia-Perera, and G. Sell, "Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays," in *Proc. CHiME-5*, 2018, pp. 6–10.
- [11] S. Meignier, "LIUM.SPDKDIARIZATION: An open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.
- [12] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Trans. on ASLP*, vol. 21, no. 10, pp. 2015–2028, 2013.
- [13] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *Proc. SLT*, 2014, pp. 413–417.
- [14] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Trans. on ASLP*, vol. 22, no. 1, pp. 217–227, 2014.
- [15] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Proc. Interspeech*, 2017, pp. 2739–2743.
- [16] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. ICASSP*, 2017, pp. 4930–4934.
- [17] M. Maciejewski, D. Snyder, V. Manohar, N. Dehak, and S. Khudanpur, "Characterizing performance of speaker diarization systems on far-field speech using standard methods," in *Proc. ICASSP*, 2018, pp. 5244–5248.
- [18] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with LSTM," in *Proc. ICASSP*, 2018, pp. 5239–5243.
- [19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on ASLP*, vol. 19, no. 4, pp. 788–798, 2011.
- [20] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *Proc. ICASSP*, 2018, pp. 4879–4883.
- [21] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.
- [22] P. A. Mansfield, Q. Wang, C. Downey, L. Wan, and I. L. Moreno, "Links: A high-dimensional online clustering method," *arXiv preprint arXiv:1801.10123*, 2018.
- [23] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Proc. Interspeech*, 2018, pp. 2808–2812.
- [24] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Žmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mošner, and P. Matějka, "BUT system for DIHARD speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.
- [25] L. Sun, J. Du, C. Jiang, X. Zhang, S. He, B. Yin, and C.-H. Lee, "Speaker diarization with enhancing speech for the first DIHARD challenge," in *Proc. Interspeech*, 2018, pp. 2793–2797.
- [26] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Proc. Interspeech*, 2019 (to appear).
- [27] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005, IJCNN 2005.

- [28] Z. Lin, M. Feng, C. Nogueira dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," in *Proc. ICLR*, 2017.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.
- [30] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. ICASSP*, 2019, pp. 5796–5800.
- [31] V. S. Narayanaswamy, J. J. Thiagarajan, H. Song, and A. Spanias, "Designing an effective metric learning pipeline for speaker diarization," in *Proc. ICASSP*, 2019, pp. 5806–5810.
- [32] V. A. Miasato Filho, D. A. Silva, and L. G. Depra Cuozzo, "Joint discriminative embedding learning, speech activity and overlap detection for the dihard speaker diarization challenge," in *Proc. Interspeech*, 2018, pp. 2818–2822.
- [33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.
- [34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.
- [35] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.
- [36] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.
- [37] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. 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.
- [38] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.
- [39] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *ICLR Workshop*, 2017.
- [40] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *Proc. ICASSP*, 2019, pp. 6301–6305.
- [41] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. CVPR*, 2018, pp. 7794–7803.
- [42] L. Ye, M. Roohan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *Proc. CVPR*, 2019, pp. 10502–10511.
- [43] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," in *Proc. Interspeech*, 2018, pp. 3723–3727.
- [44] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," *Proc. ICASSP*, pp. 5884–5888, 2018.
- [45] W. Jun and L. Shengchen, "Self-attention mechanism based system for DCASE2018 challenge task1 and task4," in *DCASE2018 Challenge*, 2018.
- [46] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2018, pp. 3573–3577.
- [47] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [48] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016, pp. 31–35.
- [49] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017, pp. 241–245.
- [50] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. on ASLP*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [51] NIST, "2000 speaker recognition evaluation plan," <https://www.nist.gov/sites/default/files/documents/2017/09/26/spk-2000-plan-v1.0.htm...pdf>, 2000.
- [52] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.
- [53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.
- [54] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," *arXiv preprints arXiv:1510.08484*, 2015.
- [55] 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 *Proc. ICASSP*, 2017, pp. 5220–5224.
- [56] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015.
- [57] NIST, "The 2009 (RT-09) rich transcription meeting recognition evaluation plan," <http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf>, 2009.