Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

DLF-EEND: Dynamic Layer Fusion for End-to-End Speaker Diarization

Bongsu Jung[1], *Wooil Kim*[1,2]

[1]Department of Computer Science and Engineering, Incheon National University,
[2]Human-centered Artificial Intelligence Research Center, Incheon National University, Incheon 22012, Republic of Korea

{qhd0330, wikim}@inu.ac.kr

Abstract

In this study, we propose Dynamic Layer Fusion for EEND (DLF-EEND), a novel approach for integrating Transformer layer information in end-to-end speaker diarization. The model introduces an auxiliary branch during training, using dynamic routing to adaptively fuse multi-resolution representations at each time step. Applying Permutation-Invariant Training (PIT) loss to the fused features enhances hierarchical learning, from low-level acoustic cues to high-level speaker separation. This preserves distinct layer-specific information and improves diarization accuracy, particularly in overlapping speech and speaker transitions. During inference, only the main branch is used, reducing computation while maintaining inter-layer fusion benefits. Experiments show DLF-EEND reduces Diarization Error Rate (DER) by 59.18% on simulated datasets and improves CALLHOME performance by 14.26%, outperforming SA-EEND.

Index Terms: speaker diarization, dynamic routing, end-to-end neural diarization, layer-wise information integration

1. Introduction

Speaker diarization determines *"who spoke when"* in an audio recording by identifying speaker activity. Traditional speaker diarization systems are primarily clustering-based and consist of multiple modules, including voice activity detection (VAD), speaker embedding extraction, and clustering. These systems first remove silence segments through VAD and extract speaker features such as x-vectors and d-vectors [1, 2]. Clustering techniques, including Agglomerative Hierarchical Clustering (AHC) and spectral clustering, are then applied to assign speaker labels [3, 4]. However, clustering-based diarization methods have several limitations, including the inability to optimize all modules jointly and challenges in handling overlapping speech [5, 6].

To address these issues, end-to-end neural diarization (EEND) was introduced, enabling direct speaker activity prediction from an audio mixture [7, 8]. Self-Attentive EEND (SA-EEND) employs a transformer-based architecture with multi-head self-attention (MHSA) to model long-range dependencies and utilizes permutation-invariant training (PIT) loss to resolve speaker-label ambiguity, thereby improving diarization performance in overlapped speech.

Recent studies have proposed various improvements to SA-EEND, focusing on auxiliary loss functions and architectural modifications [9, 10]. Yu et al. [9] observed that lower layers in deep SA-EEND models contribute less to diarization accuracy. To address this, RX-EEND [9] introduced residual connections and per-block diarization loss to ensure balanced learning across layers. Jeoung et al. [10] further refined MHSA by imposing speaker-wise constraints on attention weights, reducing redundancy in attention patterns. Additionally, self-distillation (SD) techniques have been explored to transfer high-level speaker-discriminative knowledge to lower layers [11].

Building on these insights, we propose Dynamic Layer Fusion EEND (DLF-EEND), which incorporates a Capsule Network-based Dynamic Routing mechanism [12] to selectively integrate multi-resolution information across encoder layers. Unlike prior methods that rely on static learning signal distribution, DLF-EEND dynamically adjusts each encoder's contribution based on speech context. The auxiliary branch selectively enhances acoustic or speaker-discriminative features as needed, and the fused representations are optimized using PIT loss, improving gradient propagation across layers. During inference, the auxiliary branch is removed, maintaining computational efficiency while preserving the benefits of hierarchical fusion.

2. Related work

2.1. Self-Attentive End-to-End Neural Diarization [8]

Self-Attentive End-to-End Neural Diarization (SA-EEND) formulates the speaker diarization task as a *multi-label classification* problem, predicting frame-wise speaker activity posteriors directly from an input speech sequence of length T, denoted as $X = [x_1, \ldots, x_T] \in \mathbb{R}^{T \times F}$ [7, 8]. Specifically, each frame t undergoes a linear transformation followed by layer normalization to obtain an initial embedding:

$$H^0 = \text{LayerNorm}(\text{Linear}(X)) \in \mathbb{R}^{T \times D} \quad (1)$$

where $H^0 = [h_1^0, \ldots, h_T^0]$, and $h_t^0 \in \mathbb{R}^D$. This embedding is then passed sequentially through L Transformer encoder blocks:

$$H^l = \text{EncoderBlock}_l(H^{l-1}) \in \mathbb{R}^{T \times D}, \quad l = 1, \ldots, L \quad (2)$$

Finally, the output of the last Transformer encoder layer, $H^L = [h_1^L, \ldots, h_T^L]$, is fed into a sigmoid activation function to estimate the speaker activity probabilities for S speakers, $\hat{\mathbf{y}}_t = [\hat{y}_1^t, \ldots, \hat{y}_S^t]$, at each time step t:

$$\hat{\mathbf{y}}_t = \text{Sigmoid}(\text{Linear}(h_t^L)) \in \mathbb{R}^S \quad (3)$$

$$\hat{Y}_{\text{main}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_T] \in \mathbb{R}^{T \times S} \quad (4)$$

In the training phase, SA-EEND employs Permutation-Invariant Training (PIT) Loss [7]. This loss function considers all possible permutations $\phi \in \text{perm}(S)$ of the speaker labels and selects the one that minimizes the binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{PIT}}(Y, \hat{Y}_{\text{main}}) = \frac{1}{TS} \min_{\phi \in \text{perm}(S)} \sum_{t=1}^{T} \text{BCE}(\mathbf{y}_t^{\phi}, \hat{\mathbf{y}}_t) \quad (5)$$

10.21437/Interspeech.2025-1059

where $\mathbf{y}_t^{\phi} \in \{0,1\}^S$ represents the ground-truth speaker labels reordered by ϕ.

3. Proposed Method

In this section, we propose a hierarchical fusion-based speaker diarization model that mitigates the hierarchical learning imbalance in the SA-EEND framework. To achieve this, we introduce an Auxiliary Branch that complements the Main Branch, integrating a Dynamic Layer Fusion mechanism based on the dynamic routing of capsule networks. This module adaptively fuses the outputs of Transformer encoder layers at each time step.

3.1. Dynamic Layer Fusion

In this study, we employ Dynamic Routing [12] to adaptively integrate Transformer layer outputs across time steps. The encoder outputs $H^l = [h_1^l, \ldots, h_T^l] \in \mathbb{R}^{T\times D}$ are projected to capsule vote embeddings $V^l = [v_1^l, \ldots, v_T^l] \in \mathbb{R}^{T\times(N\times D')}$, where $v_t^l \in \mathbb{R}^{N\times D'}$ is computed as:

$$v_t^l = h_t^l W^l, \quad l = 1, \ldots, L, \quad t = 1, \ldots, T \tag{6}$$

where N represents the number of high-level capsules, and D' is the capsule dimension. The matrix v_t^l consists of N vote vectors; its j-th row, denoted as $v_{j,t}^l \in \mathbb{R}^{D'}$, represents the vote from layer l at time t to the j-th high-level capsule.

To ensure that the model dynamically adjusts attention weights across layers, we apply an iterative routing process. In each iteration, the routing logits $b_{j,t}^l$ are normalized via softmax, followed by the squash operation and an agreement accumulation process, defined as follows:

$$\text{squash}(s_{j,t}) = \frac{\|s_{j,t}\|^2}{1+\|s_{j,t}\|^2} \cdot \frac{s_{j,t}}{\|s_{j,t}\|} \tag{7}$$

$$b_{j,t}^l \leftarrow b_{j,t}^l + \langle v_{j,t}^l, C_{j,t}\rangle \tag{8}$$

The squash function in Eq. (7) ensures that capsule activations are properly scaled, preventing excessive growth, while Eq. (8) updates the routing logits by accumulating agreement scores, allowing the model to assign higher weights to more confident capsules. This inner product operation accumulates agreement between the frame-wise vote embeddings and the capsule outputs. By iteratively refining the agreement among capsule embeddings, Dynamic Routing enables the network to selectively integrate useful information from each Transformer layer while suppressing redundant representations.

After R routing iterations, the final capsule embeddings $\{C_{j,t}\}_{j=1}^N$ at each time step t are concatenated and mapped back to the Transformer embedding space:

$$h_t^{\text{aux}} = \text{Linear}(\text{Concat}[C_{1,t}, \ldots, C_{N,t}]) \tag{9}$$

Thus, the final auxiliary embeddings $h_t^{\text{aux}} \in \mathbb{R}^D$ are obtained. Extending this across all frames, we derive the frame-wise dynamically fused representation $H^{\text{aux}} = [h_1^{\text{aux}}, \ldots, h_T^{\text{aux}}] \in \mathbb{R}^{T\times D}$. The detailed routing initialization and iterative structure can be found in Algorithm 1.

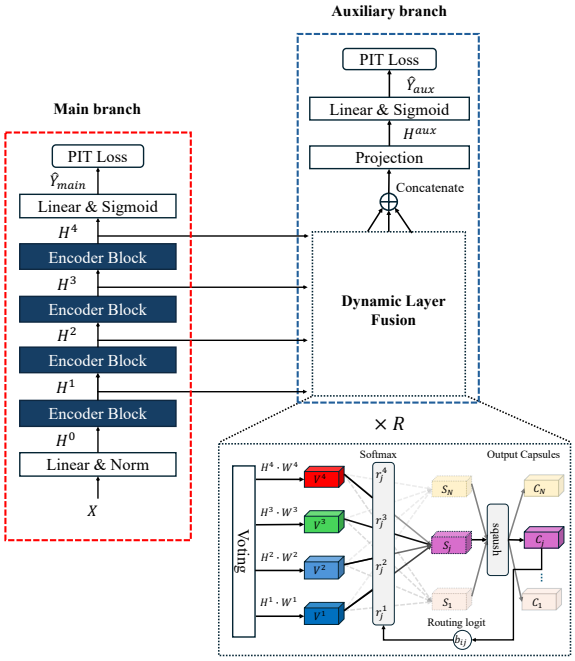


Figure 1: *Overall Architecture of DLF-EEND*

Algorithm 1: Dynamic Routing for Layer Fusion

Input: Transformer outputs $\{H^l\}_{l=1}^L$, number of capsules N, routing iterations R.
Output: Capsule embeddings $\{C_{j,t}\}_{j=1}^N$ for each time step t.
foreach $l \in \{1, \ldots, L\}$ **do**
  foreach $t \in \{1, \ldots, T\}$ **do**
    $v_t^l \leftarrow h_t^l \cdot W^l$ `// Frame-wise capsule voting`
Initialize $b_{j,t}^l \leftarrow 0, \forall l, j, t$;
for *iteration* $\leftarrow 1$ **to** R **do** `// Routing loop`
  foreach $l \in 1:L, j \in 1:N, t \in 1:T$ **do**
    $r_{j,t}^l \leftarrow \text{Softmax}_l(b_{j,t}^l)$; `// Softmax over` l
  foreach $j \in 1:N, t \in 1:T$ **do**
    $s_{j,t} \leftarrow \sum_{l=1}^L r_{j,t}^l \cdot v_{j,t}^l$;
    $C_{j,t} \leftarrow \text{squash}(s_{j,t})$;
  foreach $l \in 1:L, j \in 1:N, t \in 1:T$ **do**
    $b_{j,t}^l \leftarrow b_{j,t}^l + \langle v_{j,t}^l, C_{j,t}\rangle$ `// Accumulate agreement`
return $\{C_{j,t}\}_{j=1}^N, \forall t$

3.2. Joint Training and Inference Strategy

To optimize diarization performance, the dynamically fused representation H^{aux} undergoes a linear transformation followed by a sigmoid activation to estimate frame-wise speaker activity:

$$\hat{\mathbf{y}}_t^{\text{aux}} = \text{Sigmoid}(\text{Linear}(h_t^{\text{aux}})) \in \mathbb{R}^S \tag{10}$$

The auxiliary loss $\mathcal{L}_{\text{aux}}$ is computed using PIT loss:

$$\mathcal{L}_{\text{aux}} = \mathcal{L}_{\text{PIT}}(Y, \hat{Y}_{\text{aux}}) \tag{11}$$

The total loss is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \lambda \mathcal{L}_{\text{aux}} \tag{12}$$

where $\mathcal{L}_{\text{main}}$ follows the PIT framework as defined in Eq. (5), and λ balances the auxiliary loss contribution. Empirical

Table 1: *Dataset Statistics for Training and Evaluation*

Dataset	Sim2spk		CALLHOME	
	Train	Test	Adapt	Test
# Mixtures	100,000	500	155	148
Overlap Ratio (%)	34.4	34.4	14.0	13.1
Noise + RIR	✓	✓	—	—

results show that $\lambda = 1$ improves gradient flow and mitigates hierarchical learning imbalance.

During inference, only the main branch is retained, ensuring efficiency while preserving the hierarchical integration benefits. The auxiliary branch, used for training, is discarded to reduce complexity without compromising speaker discrimination.

4. Experiments

All experiments were conducted on an Ubuntu 20.04 system with PyTorch 1.12.1 and a single NVIDIA GeForce RTX 4090 GPU. The proposed model was implemented using the publicly available SA-EEND PyTorch repository[1].

4.1. Datasets

We used both simulated and real-world datasets. The simulated dataset, Sim2spk, was generated using the speech mixture simulation algorithm from [8][2], based on the LibriSpeech corpus [13]. Each mixture contains speech from two different speakers, with 10 to 20 utterances randomly sampled per speaker. To match the real dataset, all simulated speech was downsampled to 8kHz before mixing. A total of 100,000 mixtures were created, maintaining an overlap ratio of 34.4% in both training and test sets. Background noise samples were drawn from the MUSAN corpus [14], and room impulse responses (RIRs) were applied with a probability of 50% [8].

For real-world evaluation, we used the CALLHOME (CH) dataset [15], which contains multi-speaker telephone conversations. Following [8], only two-speaker utterances were extracted for this study. Part 1 was used for model adaptation, while Part 2 was used for evaluation. Table 1 summarizes the dataset statistics.

4.2. Experiment Setup

For fair comparison, all SA-EEND-based models, including the proposed DLF-EEND, share the same architectural configuration. The Transformer encoder consists of $L = 4$ blocks, each with $H = 4$ attention heads and an attention dimension of $D = 256$. The Feed-Forward Network (FFN) has $D_{\text{FFN}} = 1024$ internal units and a dropout rate of 0.1. Input features are 23-dimensional log-mel filterbank energies, extracted using a 25ms frame length and 10ms frame shift. Each frame was spliced with 7 left and 7 right context frames, forming 15-frame segments. The sequence was processed with a stride of 10 frames. The Dynamic Routing module in DLF-EEND uses a capsule dimension of $D' = 256$. Routing iterations were set to $R = 3$ for training and $R = 2$ for adaptation. All models were pretrained on the Sim2spk dataset and subsequently adapted to CALLHOME. The Adam optimizer [16] was used with a Noam learning rate scheduler [17], applying 25K warm-up steps. Both phases ran for 100 epochs, and final model parameters were obtained by averaging those from the last 10 epochs. For evaluation, the Diarization Error Rate (DER) [18] was computed with a collar tolerance of 0.25s, following [8]. An 11-frame median filter and a fixed 0.5 threshold were applied for post-processing.

[1] https://github.com/Xflick/EEND_PyTorch.git

[2] https://github.com/hitachi-speech/EEND

Table 2: *Effect of Capsule Count on DER (%)*

Capsule Count	Sim2spk	CALLHOME
SA-EEND (Baseline)	4.90	13.46
1 Capsule	3.82	12.08
2 Capsules	2.68	11.89
3 Capsules	2.45	11.97
4 Capsules	**2.00**	**11.54**
5 Capsules	2.70	12.60

Table 3: *DER(%) Performance Comparison of SA-EEND, RX-EEND, and DLF-EEND*

Method	Sim2spk				CALLHOME			
	MI	FA	CF	DER	MI	FA	CF	DER
SA-EEND (Baseline)	1.74	2.97	0.19	4.90	7.78	2.87	2.81	13.46
RX-EEND	1.26	1.84	**0.16**	3.26	7.14	**2.17**	3.03	12.34
DLF-EEND (Proposed)	**0.88**	**0.93**	0.20	**2.00**	**6.40**	2.69	**2.45**	**11.54**

5. Results and Analysis

5.1. Impact of Capsule Count on DER Performance

Table 2 presents the DER performance as a function of capsule count. The optimal performance was achieved with four capsules, yielding the lowest DERs on both datasets (Sim2spk: 2.00%, CALLHOME: 11.54%). A single capsule resulted in higher DERs, indicating that inadequate layer fusion limits feature integration. In contrast, using five capsules degraded performance, suggesting that an excessive number of capsules may lead to redundant or noisy representations. These findings underscore the importance of selecting an optimal capsule count for robust speaker diarization.

5.2. Comparison of DER Performance

Table 3 presents the Diarization Error Rate (DER) of SA-EEND [8], RX-EEND [9], and the proposed DLF-EEND model. Results for RX-EEND were reproduced under the same experimental conditions for fair comparison. DLF-EEND achieved the lowest DER in both datasets, demonstrating superior performance over SA-EEND and RX-EEND. Particularly, in the Simulated dataset with a high proportion of overlapping speech, DLF-EEND significantly reduced Missed Detection (MI) and False Alarm (FA) rates, highlighting the effectiveness of dynamic routing in adapting to varying speech conditions.

5.3. Analysis of Capsule Weights in Dynamic Routing

Figure 2 presents the capsule weight distribution across Transformer layers in Dynamic Routing, illustrating how different layers contribute to diarization. (a), (b) show the frame-wise averaged capsule weights, min-max normalized to [0,1], while

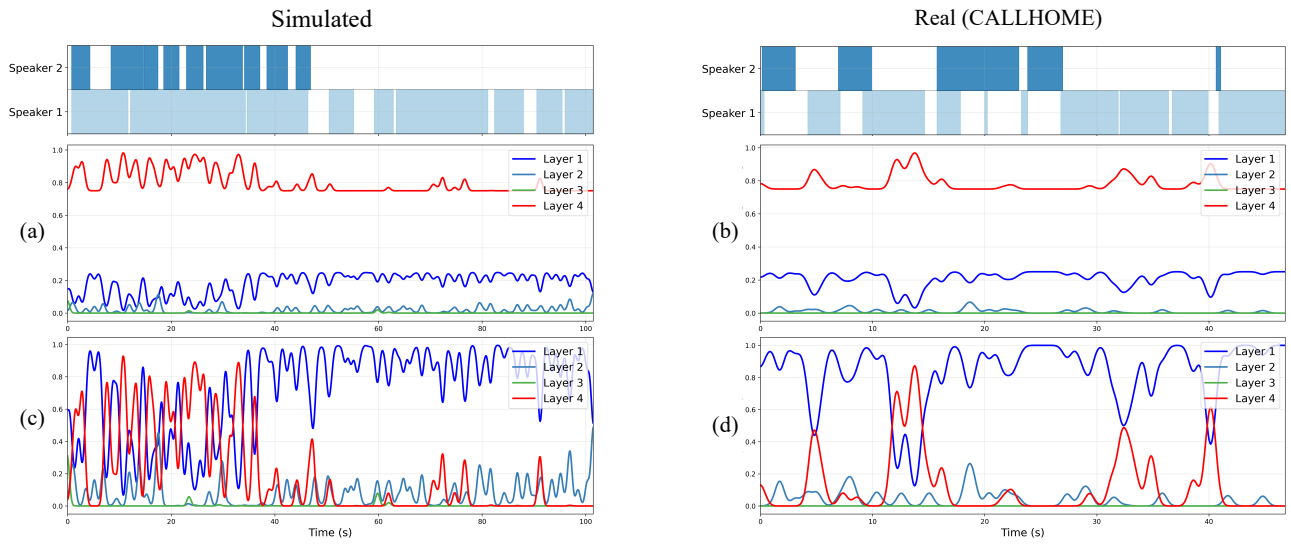


Figure 2: *Visualization of capsule weight distributions in Dynamic Routing. (a), (b) Normalized average weights of four capsules for simulated and CALLHOME datasets (CALLHOME recording ID: iaaq), where the weights for all capsules were first averaged at each frame and then min-max normalized to the range [0,1]. (c), (d) Frame-wise weight distribution of the second capsule.*

(c), (d) visualize the weight distribution of the second capsule. As shown in Figure 2(a) and (b), Layer 4 exhibits the highest activations across both datasets, aligning with its role in capturing speaker-discriminative features. This suggests that deeper Transformer layers primarily focus on high-level speaker representations. Notably, Layer 4 maintains strong activations in overlapping speech, indicating its effectiveness in distinguishing multiple speakers. In contrast, Layers 2 and 3 contribute less, implying that intermediate layers are less involved in direct speaker separation.

Figures 2(c) and (d) further reveal that Capsule 2 assigns higher weights to Layer 1, particularly in non-overlapping speech segments. This suggests that lower layers capture essential acoustic features such as speech continuity and background characteristics. In speaker transitions, Layer 4 activations increase, reflecting its role in distinguishing new speakers, whereas lower layers become dominant in extended single-speaker segments, where acoustic continuity is more relevant than explicit speaker separation.

These findings confirm that Dynamic Routing adaptively adjusts layer contributions based on speech context. By selectively integrating multi-resolution information, DLF-EEND effectively utilizes both high-level speaker-discriminative features and low-level acoustic cues, enhancing diarization accuracy in complex speech scenarios.

5.4. Impact of Layer Fusion in the Auxiliary Branch

Table 4 evaluates different layer fusion strategies in the Auxiliary Branch. Static fusion methods, including Concatenation, Weighted Sum, and Simple Average, failed to capture temporal speaker variations, resulting in higher DERs.

Notably, the Weighted Sum method, which applies learnable weights to each Transformer layer output and then sums them, exhibited high DERs (8.80% on Sim2spk, 16.73% on CALLHOME), suggesting that fixed-weight combinations lead to suboptimal learning.

In contrast, Dynamic Routing significantly improved performance, achieving DERs of 1.97% (Sim2spk) and 11.53% (CALLHOME), demonstrating its ability to adaptively integrate layer-wise speaker-discriminative features. Even after the auxiliary branch was removed during inference, the main branch maintained a competitive DER of 2.00% (Sim2spk) and 11.54% (CALLHOME), confirming the robustness of hierarchical fusion.

Table 4: *Impact of Layer Fusion Strategies in the Auxiliary Branch (DER, %)*

Method	**Auxiliary Branch DER**		**Main Branch DER**	
	Sim2spk	CALLHOME	Sim2spk	CALLHOME
(SA-EEND)	-	-	4.90	13.46
Concatenation	6.09	16.78	6.29	14.10
Weighted Sum	8.80	16.73	7.69	13.66
Simple Average	4.07	14.58	4.45	12.28
Dynamic Routing	**1.97**	**11.53**	**2.00**	**11.54**

6. Conclusion

In this study, we proposed Dynamic Layer Fusion EEND (DLF-EEND), which employs dynamic routing to integrate multi-resolution speaker representations across Transformer layers. DLF-EEND adaptively adjusts layer contributions based on speech context, effectively capturing both high- level speaker-discriminative features and low-level acoustic features. Experimental results demonstrated that DLF-EEND significantly reduces DER in both overlapping speech and speaker transition scenarios, outperforming existing EEND approaches. Additionally, the capsule count was identified as a key factor influencing performance, highlighting the importance of optimizing the layer fusion mechanism. This study was conducted under a two-speaker setting, and future work will explore its scalability to multi-speaker scenarios. Furthermore, we aim to enhance adaptive weight learning to improve model robustness while reducing hyperparameter sensitivity and computational overhead.

7. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

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

[3] D. Garcia-Romero, D. Snyder, D. Povey, and S. Khudanpur, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[4] G. Sell and D. Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4794–4798.

[5] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[6] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Cernocky, "Bayesian hmm based x-vector clustering for speaker diarization," in *Interspeech*, 2019.

[7] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," in *Interspeech 2019*, 2019, pp. 4300–4304.

[8] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[9] Y. Yu, D. Park, and H. K. Kim, "Auxiliary loss of transformer with residual connection for end-to-end speaker diarization," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8377–8381.

[10] Y.-R. Jeoung, J.-Y. Yang, J.-H. Choi, and J.-H. Chang, "Improving transformer-based end-to-end speaker diarization by assigning auxiliary losses to attention heads," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[11] Y.-R. Jeoung, J.-H. Choi, J.-S. Seong, J. Kyung, and J.-H. Chang, "Self-distillation into self-attention heads for improving transformer-based end-to-end neural speaker diarization," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2023. International Speech Communication Association, 2023, pp. 3197–3201.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[13] V. Panayotov, G. Chen, D. Povey, and S. 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.

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

[15] M. Przybocki and A. Martin, "2000 nist speaker recognition evaluation," in *Proceedings of the Speaker Recognition Workshop*. Philadelphia: Linguistic Data Consortium, 2001.

[16] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[18] J. G. Fiscus, J. Ajot, and J. S. Garofolo, "The rich transcription 2007 meeting recognition evaluation," in *International Evaluation Workshop on Rich Transcription*. Springer, 2007, pp. 373–389.